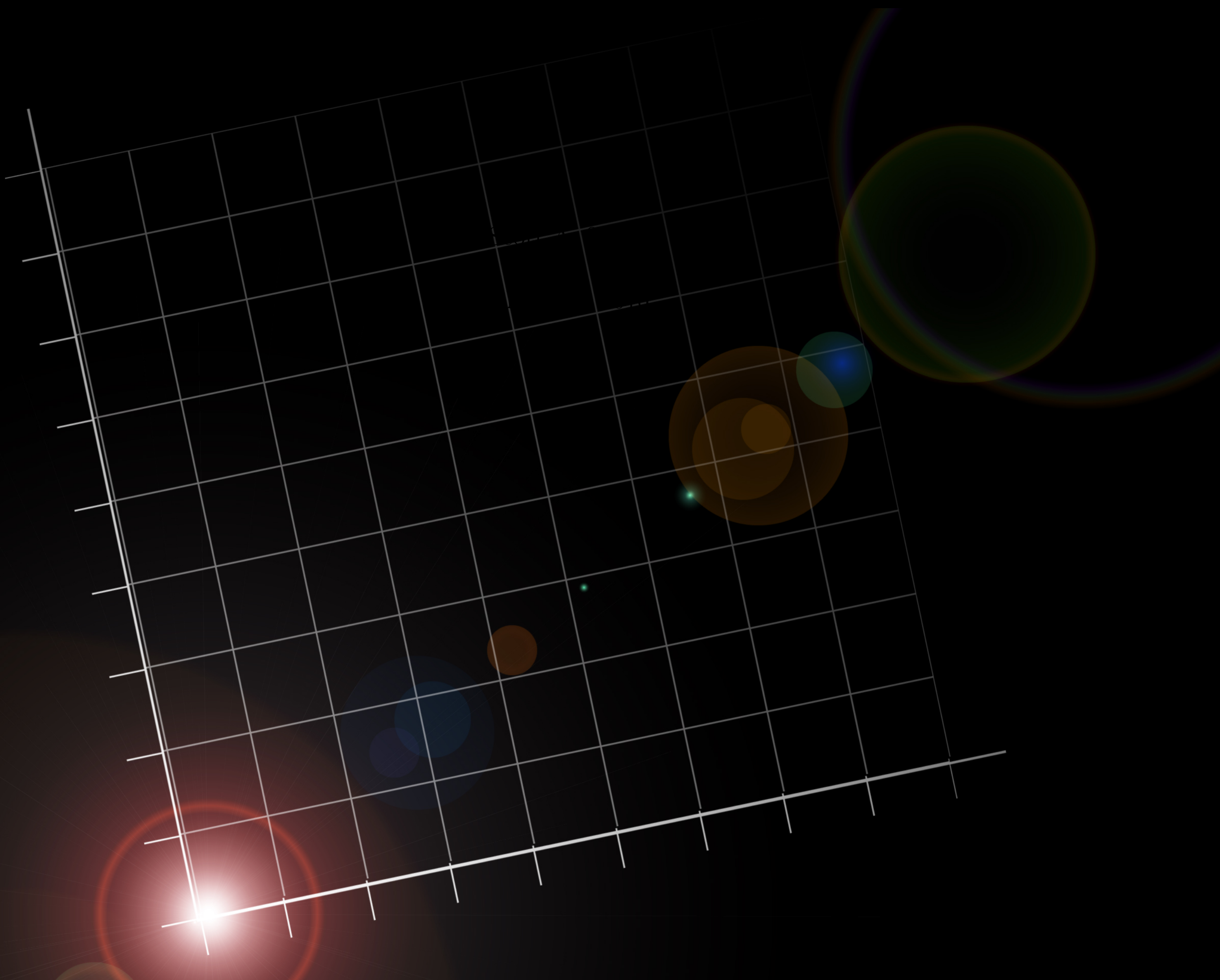
Matrices, Vectors, and 3D Math

A Game Programming Approach with MATLAB®



Scott Stevens

Matrices, Vectors, and 3D Math

A Game Programming Approach with MATLAB®

Scott Stevens

www.StevensMV3D.com

Matrices, Vectors, and 3D Math
A Game Programming Approach with MATLAB®

v. 05192016
www.centerofmath.org

ISBN-10: 0-9842071-8-X
ISBN-13: 978-0-9842071-8-3

Contents

Preface

This textbook originates from the class notes used in a math course developed at Champlain College for Software Engineer and Game Programming majors. I wrote this book because I could not find a suitable text which satisfied the following criteria.

- The course covers basic topics from Linear Algebra and Calculus III.
- The course can be completed in one semester.
- The course has a single semester of college-level calculus as a prerequisite.
- The course is suited for Software Engineer and Game Programming majors as well as a variety of other students interested in mathematics.

Creating such a course/text was not an easy task. I am, after all, a mathematician and was accustomed to teaching the standard sequence of 4-credit Calculus courses and then Linear Algebra where everything fits nicely where it belongs with minimal gaps in the underlying knowledge structure. However, the students taking this course did not have that kind of credit allowance in their curriculum. So I tried to create a course which was founded on a solid mathematical structure but allowed students to start doing interesting and moderately sophisticated mathematics in a short period of time.

As it turned out, the field of game-programming provided the perfect back-drop for such a course. What started out as a math course for game-programmers quickly, and quite naturally, became a sequence of game programming objectives that motivated the very topics I wanted to cover in mathematics. After having taught this course many times I am now of the belief that more math courses and topics should be motivated by such objectives. The reason is simple. It does not take long to describe the objective in a programmable game. There are many examples. You want the ball (or object or figure) to move around in a realistic manner. You want to know if it has hit anything else in its world. You want to determine an appropriate response to such a collision. You want to graphically represent these objects and reactions in a visually realistic way. These relatively simple objectives lead, quite naturally, to the development of many of the topics in mathematics often motivated by less visible and more abstract applications. In this course the bigger objectives are less like *determine where the projectile hits the ground* and more like *animate the entire trajectory of the projectile for the first 10 seconds of flight including bounces.* Obviously the second objective entails the first but covers so much more and, in the end, produces an answer that the students can not only visualize but actually watch on the computer screen.

Obviously, programmable software is required for a course such as this. There are many options and we considered quite a few before choosing MATLAB® . There are *game engines*, *physics engines*, and game

development software. These options (while quite appealing to many of the students) were dropped as these software packages generally do most, if not all, of the math for you. This is a math course and that would defeat the purpose. There are a few suitable mathematical software packages that were considered. We eventually chose MATLAB® because, in my opinion, programming in MATLAB® is very simple, the graphics are superb, animations are easily created, and the software is reasonably priced. However, there are plenty of people who would prefer other packages. Unfortunately, we could only pick one, so we chose MATLAB® . Perhaps future versions of this text will be written for other packages.

As with any motivating applications in a math course, some time must be dedicated to teaching/learning the background information. In this case, that background comes in the form of programming, graphing, and making animations in MATLAB® . There is a brief but fairly thorough introduction to MATLAB® in the appendix. This appendix is complete with an assignment that, once completed, should have the students fairly comfortable with the software and how to use it for the purposes of this course. This is usually completed in less than two class meetings. In order to expedite this part of the learning curve, all of the code used for demonstration purposes, and some problem set *starter code* can be downloaded from the textbook companion website found at

www.StevensMV3D.com.

Many of the problems and projects within the text can be completed by downloading the appropriate demonstration or *starter* code and making minimal yet significant alterations. I.e., Students should not have to write much code from scratch.

While the students who use this book should have completed a college-level calculus course, that does not always ensure they have a good grasp of trigonometry. For this reason, there is also an appendix on trigonometry, complete with a summary problem set. Additionally, the Differentiation Appendix provides a few tables of standard differentiation rules.

The course has been a pleasure to develop and teach. This book has been a necessary part of that process. I now appreciate Game Programming for its bountiful supply of applied math problems and suspect that it will play a larger role in motivating math students in the future. I hope you enjoy teaching or learning mathematics with the aid of this book. It was fun to write. As I tell my students quite often - *I wish I had taken this course when I was in college.* I really do.

Scott Stevens

Chapter 1

Introduction to Linear Algebra

Linear Algebra is a branch of mathematics concerned with the study of linear systems, vectors, vector spaces, matrices, transformations, and much more. It can be presented in a very abstract manner which includes the algebraic properties of vector spaces and subspaces. It has extensive applications in the natural and social sciences. However, we will restrict our study of linear algebra to include only the most basic applications which we will need in studying systems of equations and the geometry of two- and three-dimensional space.

1.1 Systems of Linear Equations

Often in mathematics we are concerned with solving equations. Specifically, if we want to solve an equation for a particular variable we are guaranteed to be able to do it provided the equation is linear in that variable. For example, consider the equation

$$x^5 + \sin x - 72y = 105 - 12y. \tag{1.1}$$

This equation is linear in y and it would be easy to solve this equation for y. However, the equation is not linear in x and hence solving this equation for x would be far more difficult and, in this case, impossible to solve algebraically. In this chapter, we will focus on solving equations for variables that appear linearly but we will be solving more than one equation at a time. When you have more than one equation to solve, the collection of equations is called a **system of equations**.

The task of solving systems of linear equations is more complicated than you might first think. For example, consider solving the single equation $ax = b$ for x. If $a \neq 0$, then $x = b/a$. However, if $a = 0$ there are two options. If $a = 0$ and $b = 0$ there are infinitely many solutions. If $a = 0$ and $b \neq 0$ there are no solutions. This same situation occurs with systems of linear equations but it is harder tell if there is one, infinitely many, or no solutions. For example, solve the following systems of equations for x and y.

(1) $\begin{array}{rcrcr} x & + & y & = & 4 \\ 3x & + & y & = & 4 \end{array}$ (2) $\begin{array}{rcrcr} x & + & y & = & 4 \\ 3x & + & 3y & = & 12 \end{array}$ (3) $\begin{array}{rcrcr} x & + & y & = & 4 \\ 3x & + & 3y & = & 10 \end{array}$

Answers: (1) $x = 0$, $y = 4$, (2) infinitely many solutions, (3) no solutions. Think about determining where two lines intersect. They can intersect at a point, infinitely many points (same line), or not at all (parallel).

There are many ways you could have solved the previous systems of equations. Some ways are better than others depending on the equations involved. In this chapter we formalize our approach to this type of problem by using a standard method which will produce a solution every time one exists. This method is called **Gaussian elimination**. The previous systems of equations were *two-by-two* systems. This means there were two equations and two unknowns (x and y). Gaussian elimination can be used on these and larger systems as well. Since we don't want to run out of letters, we will often order our variables with subindexes. First, create the **augmented matrix** representing the system of equations. This consists of the **coefficient matrix**, a vertical line, and the column of constants.

System of Equations

$$\begin{array}{rcrcrcr} 2x_1 & + & 4x_2 & + & 6x_3 & = & 18 \\ 4x_1 & + & 5x_2 & + & 6x_3 & = & 24 \\ 3x_1 & + & x_2 & - & 2x_3 & = & 4 \end{array}$$

Augmented Matrix

$$\left[\begin{array}{ccc|c} 2 & 4 & 6 & 18 \\ 4 & 5 & 6 & 24 \\ 3 & 1 & -2 & 4 \end{array}\right]$$

We will perform Gaussian elimination on the augmented matrix. The goal is to perform various row operations which result in equivalent systems until the coefficient matrix has one's on the diagonal and zeros below the diagonal.

The Idea Behind Gaussian Elimination

1. Work from the top left to the bottom right of the coefficient matrix.
2. At each column get a 1 on the diagonal and all zeros below it.
3. Continue this and try to get 1's along the diagonal and zeros below it.
4. This is called **row-echelon form.**

$$\left[\begin{array}{cccc|c} 1 & * & * & * & * \\ 0 & 1 & * & * & * \\ 0 & 0 & 1 & * & * \\ 0 & 0 & 0 & 1 & * \end{array}\right]$$

Once the augmented matrix is in row echelon form, we can use **back substitution** to solve for the variables. We'll get to this later. There are only three types of row operations required to get the augmented matrix into row echelon form.

ROW OPERATIONS [notation]:

1. Multiply a row by a number. $[R_i \to a\ R_i]$
2. Add/subtract a multiple of one row to/from another and replace it. $[R_i \to R_i \pm aR_j]$
3. Switch any two rows. $[R_i \leftrightarrow R_j]$

• Example 1

System of Equations	Augmented Matrix	Row Operation(s)
$\begin{array}{rcrcrcr} 2x_1 & + & 4x_2 & + & 6x_3 & = & 18 \\ 4x_1 & + & 5x_2 & + & 6x_3 & = & 24 \\ 3x_1 & + & x_2 & - & 2x_3 & = & 4 \end{array}$	$\left[\begin{array}{ccc\|c} 2 & 4 & 6 & 18 \\ 4 & 5 & 6 & 24 \\ 3 & 1 & -2 & 4 \end{array}\right]$	$R_1 \to 1/2\ R_1$
$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 9 \\ 4x_1 & + & 5x_2 & + & 6x_3 & = & 24 \\ 3x_1 & + & x_2 & - & 2x_3 & = & 4 \end{array}$	$\left[\begin{array}{ccc\|c} 1 & 2 & 3 & 9 \\ 4 & 5 & 6 & 24 \\ 3 & 1 & -2 & 4 \end{array}\right]$	$R_2 \to R_2 - 4R_1$ $R_3 \to R_3 - 3R_1$
$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 9 \\ & & -3x_2 & - & 6x_3 & = & -12 \\ & & -5x_2 & - & 11x_3 & = & -23 \end{array}$	$\left[\begin{array}{ccc\|c} 1 & 2 & 3 & 9 \\ 0 & -3 & -6 & -12 \\ 0 & -5 & -11 & -23 \end{array}\right]$	$R_2 \to -1/3\ R_2$
$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 9 \\ & & x_2 & + & 2x_3 & = & 4 \\ & & -5x_2 & - & 11x_3 & = & -23 \end{array}$	$\left[\begin{array}{ccc\|c} 1 & 2 & 3 & 9 \\ 0 & 1 & 2 & 4 \\ 0 & -5 & -11 & -23 \end{array}\right]$	$R_3 \to R_3 - (-5)\ R_2$
$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 9 \\ & & x_2 & + & 2x_3 & = & 4 \\ & & & - & 1x_3 & = & -3 \end{array}$	$\left[\begin{array}{ccc\|c} 1 & 2 & 3 & 9 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & -1 & -3 \end{array}\right]$	$R_3 \to -1\ R_3$
$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 9 \\ & & x_2 & + & 2x_3 & = & 4 \\ & & & & x_3 & = & 3 \end{array}$	$\left[\begin{array}{ccc\|c} 1 & 2 & 3 & 9 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 1 & 3 \end{array}\right]$	This is **Row Echelon Form**

Gaussian Elimination Stops Here.
You solve for the variables using **back substitution**. This means you start at the last equation and solve for the last variable and work your way to the first equation substituting the values you find along the way.

- The third row of the augmented matrix represents the equation $\mathbf{x_3 = 3}$.
- The second row of the augmented matrix represents the equation $x_2 + 2x_3 = 4$ but we have $x_3 = 3$.

$$x_2 + 2x_3 = 4 \quad \to \quad x_2 + 6 = 4 \quad \to \quad \mathbf{x_2 = -2}$$

- The first row of the augmented matrix represents $x_1 + 2x_2 + 3x_3 = 9$ but we have x_2 and x_3.

$$x_1 + 2x_2 + 3x_3 = 9 \quad \to \quad x_1 - 4 + 9 = 9 \quad \to \quad \mathbf{x_1 = 4}$$

- The solution is $x_1 = 4$, $x_2 = -2$, $x_3 = 3$.

You should go back and check that these values satisfy each equation of the original system.

• Example 2

System of Equations	Augmented Matrix	Row Operation(s)
$\begin{aligned} -3x_1 - 8x_2 - 8x_3 &= -7 \\ 2x_1 + 6x_2 + 10x_3 &= 9 \\ 1x_1 + 3x_2 + 4x_3 &= 3 \end{aligned}$	$\left[\begin{array}{ccc\|c} -3 & -8 & -8 & -7 \\ 2 & 6 & 10 & 9 \\ 1 & 3 & 4 & 3 \end{array}\right]$	$R_1 \leftrightarrow R_3$
$\begin{aligned} 1x_1 + 3x_2 + 4x_3 &= 3 \\ 2x_1 + 6x_2 + 10x_3 &= 9 \\ -3x_1 - 8x_2 - 8x_3 &= -7 \end{aligned}$	$\left[\begin{array}{ccc\|c} 1 & 3 & 4 & 3 \\ 2 & 6 & 10 & 9 \\ -3 & -8 & -8 & -7 \end{array}\right]$	$R_2 \to R_2 - 2R_1$ $R_3 \to R_3 + 3R_1$
$\begin{aligned} 1x_1 + 3x_2 + 4x_3 &= 3 \\ 2x_3 &= 3 \\ x_2 + 4x_3 &= 2 \end{aligned}$	$\left[\begin{array}{ccc\|c} 1 & 3 & 4 & 3 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 4 & 2 \end{array}\right]$	$R_2 \leftrightarrow R_3$
$\begin{aligned} 1x_1 + 3x_2 + 4x_3 &= 3 \\ x_2 + 4x_3 &= 2 \\ 2x_3 &= 3 \end{aligned}$	$\left[\begin{array}{ccc\|c} 1 & 3 & 4 & 3 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 2 & 3 \end{array}\right]$	$R_3 \to 1/2\ R_3$
$\begin{aligned} 1x_1 + 3x_2 + 4x_3 &= 3 \\ x_2 + 4x_3 &= 2 \\ x_3 &= 3/2 \end{aligned}$	$\left[\begin{array}{ccc\|c} 1 & 3 & 4 & 3 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 1 & 3/2 \end{array}\right]$	This is **Row Echelon Form**

Gaussian Elimination Stops Here. You solve for the variables using **back substitution**.

- The third equation is $\mathbf{x_3 = 3/2}$.
- The second equation is $x_2 + 4x_3 = 2 \quad \to \quad x_2 + 6 = 2 \quad \to \quad \mathbf{x_2 = -4}$.
- The first equation is $x_1 + 3x_2 + 4x_3 = 3 \quad \to \quad x_1 - 12 + 6 = 3 \quad \to \quad \mathbf{x_1 = 9}$.
- The solution is $x_1 = 9$, $x_2 = -4$, $x_3 = 3/2$.

From now on, the preceding sequence of row operations will be denoted simply by

$$\left[\begin{array}{ccc|c} -3 & -8 & -8 & -7 \\ 2 & 6 & 10 & 9 \\ 1 & 3 & 4 & 3 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & 4 & 3 \\ 2 & 6 & 10 & 9 \\ -3 & -8 & -8 & -7 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & 4 & 3 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 4 & 2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & 4 & 3 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 2 & 3 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & 4 & 3 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 1 & 3/2 \end{array}\right].$$

It is up to you to properly infer the associated row operations. If you can get one matrix from another by performing any number of sequential row operations, the matrices are called **row equivalent**. We use the symbol $\sim$ to represent row equivalence.

In both of the previous examples we were lucky. Why?

1. We didn't have to deal with fractions. In most problems there will inevitably be more complicated calculations. In the future, we will be using a computer to perform Gaussian elimination. For now, the problems won't get too messy.

2. We were dealing with a square system of equations. This means the number of equations is the same as the number of variables. The coefficient portion of the augmented matrix is square. It is the most common type of system of equations. This represents our best shot at getting a unique solution. We'll deal with non-square systems later.

3. We were able to obtain row echelon form through a sequence of row operations. As such, we were able to obtain a unique solution. This doesn't always happen. When it doesn't we can get different solution options. This is especially frequent when dealing with non-square systems but it can happen with square systems as well.

Types of Systems and Types of Solutions

When dealing with systems of equations, there are three types of solutions. There can be a unique solution, infinitely many solutions, or no solution. The type of system determines the type of solution. These types are outlined below.

1. A **consistent system** has at least one solution and there are two options.

   (a) The system has a unique solution. Such a system is called an **independent** system.

   (b) The system has infinitely many solutions. Such a system is called a **dependent** system.

2. An **inconsistent** system has no solutions.

The first two examples of this section resulted in a unique solution. Both of those systems were consistent and independent. The next two examples demonstrate how to deal with systems that do not have a unique solution. If there are infinitely many solutions, we need a way to describe how such solutions are structured. We do this by defining a general solution. If there are no solutions, we need to be able to make this determination at some point in the solution process. At first inspection, there is no easy way to distinguish the different types of systems. Future sections in this chapter reveal some matrix properties that will help with this task. For now, the only way to determine the type of system is to proceed through the solution process as demonstrated in the next two examples.

- **Example 3: Infinite Number of Solutions → Consistent and Dependent System**

$$\begin{array}{ccccccc} 2x_1 & + & 4x_2 & + & 6x_3 & = & 18 \\ 4x_1 & + & 5x_2 & + & 6x_3 & = & 24 \\ 2x_1 & + & 7x_2 & + & 12x_3 & = & 30 \end{array}$$

$$\left[\begin{array}{ccc|c} 2 & 4 & 6 & 18 \\ 4 & 5 & 6 & 24 \\ 2 & 7 & 12 & 30 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 9 \\ 4 & 5 & 6 & 24 \\ 2 & 7 & 12 & 30 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 9 \\ 0 & -3 & -6 & -12 \\ 0 & 3 & 6 & 12 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 9 \\ 0 & 1 & 2 & 4 \\ 0 & 3 & 6 & 12 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 9 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Here, the third equation has essentially disappeared - it is meaningless: $0x_1 + 0x_2 + 0x_3 = 0$. This has infinitely many solutions. So, we turn to the second equation: $x_2 + 2x_3 = 4$ and let x_3 be a **free variable**. This is done by setting $x_3 = t$ where t represents any real number.

Let $x_3 = t$

From the second equation:

$x_2 + 2x_3 = 4$
$x_2 = 4 - 2x_3$
$x_2 = 4 - 2t$

From the first equation:

$x_1 + 2x_2 + 3x_3 = 9$
$x_1 = 9 - 2x_2 - 3x_3$
$x_1 = 9 - 2(4 - 2t) - 3t$
$x_1 = 9 - 8 + 4t - 3t$
$x_1 = 1 + t$

The **general solution** is given in the form

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1+t \\ 4-2t \\ t \end{bmatrix} \quad \text{for} \quad -\infty < t < \infty.$$

This actually represents a line in 3 space. A **particular solution** is found by assigning any number to the parameter t. For example, if we set $t = 0$, a particular solution is $x_1 = 1$, $x_2 = 4$, and $x_3 = 0$.

- **Example 4: No Solutions → An inconsistent system**

$$\begin{array}{ccccccc} & & x_2 & - & 2x_3 & = & 3 \\ x_1 & - & 2x_2 & + & 5x_3 & = & 10 \\ 2x_1 & - & 8x_2 & + & 18x_3 & = & 11 \end{array}$$

$$\left[\begin{array}{ccc|c} 0 & 1 & -2 & 3 \\ 1 & -2 & 5 & 10 \\ 2 & -8 & 18 & 11 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & -2 & 5 & 10 \\ 0 & 1 & -2 & 3 \\ 2 & -8 & 18 & 11 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & -2 & 5 & 10 \\ 0 & 1 & -2 & 3 \\ 0 & -4 & 8 & -9 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & -2 & 5 & 10 \\ 0 & 1 & -2 & 3 \\ 0 & 0 & 0 & 3 \end{array}\right].$$

The third equation says $0x_1 + 0x_2 + 0x_3 = 3$. This equation has no solutions and so the system of equations has no solution and we call the system **inconsistent**.

Chapter 1.1 Worksheet

Use Gaussian elimination with back-substitution to determine if the system of equations is consistent (has at least one solution). If it is consistent, solve for the variables. If you get infinitely many solutions, give the **general solution** in terms of a parameter (t) and give one **particular solution**.

1.
$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

2.
$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

3.
$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 10 \\ x_1 + 5x_2 &= 6 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

Chapter 1.1 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. At the very beginning of this section we noticed that solving 2 equations in 2 unknowns was like determining if and how two lines can intersect in a plane. Now suppose we have 3 planes (geometric plane, not a flying machine) in three dimensional space. In what ways can they intersect?

2. Explicitly write out the system of equations represented by the given augmented matrix. Use the variables (x, y, z) if there are 3 or fewer variables and $(x_1, x_2, x_3, x_4, \ldots)$ if there are four or more variables.

   (a)* $\left[\begin{array}{ccc|c} 2 & 4 & 6 & 18 \\ 4 & 0 & -6 & 24 \\ 0 & 1 & -2 & 4 \end{array}\right]$ (b) $\left[\begin{array}{cccc|c} -2 & 1 & 3 & -4 & 10 \\ 0 & 0 & 3 & 4 & 9 \\ 1 & 0 & -2 & -3 & 0 \end{array}\right]$

3. Express the given system of equations in terms of an augmented matrix.

   (a)* $\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= 22 \\ 3x_2 - x_3 - 4x_4 &= -2 \\ x_1 - 2x_2 - 3x_3 &= 7 \end{aligned}$ (b) $\begin{aligned} x - y + z &= 1 \\ y - 6z &= 2 \\ x - 6z &= 5 \end{aligned}$

Probems 4 - 7: Use Gaussian elimination with back-substitution to determine if the system of equations is consistent (has at least one solution) or not. If it is consistent, solve for the variables. If you get infinitely many solutions, give the **general solution** in terms of a parameter (t) and give one **particular solution**.

4.* (a) $\begin{aligned} x_1 + 3x_2 - x_3 &= 1 \\ x_2 - x_3 &= 1 \\ 2x_1 - x_2 + 5x_3 &= 2 \end{aligned}$ (b) $\begin{aligned} x_1 + 3x_2 - x_3 &= 1 \\ x_2 - x_3 &= 1 \\ 2x_1 - x_2 + 5x_3 &= -5 \end{aligned}$ (c) $\begin{aligned} x_1 - 3x_3 &= -5 \\ 2x_1 + x_2 + 2x_3 &= 7 \\ 3x_1 + 2x_2 + x_3 &= 7 \end{aligned}$

5. (a) $\begin{aligned} x_1 + x_2 + 2x_3 &= 8 \\ -x_1 - 2x_2 + 3x_3 &= 1 \\ 3x_1 - 7x_2 + 4x_3 &= 10 \end{aligned}$ (b) $\begin{aligned} x_1 + 2x_2 &= 0 \\ 2x_1 + x_2 + 3x_3 &= 0 \\ -2x_2 + 2x_3 &= 0 \end{aligned}$ (c) $\begin{aligned} x_1 + 2x_2 &= 1 \\ 2x_1 + x_2 + 3x_3 &= 2 \\ -2x_2 + 2x_3 &= 3 \end{aligned}$

6.* (a) $\begin{aligned} x + 2y + 3z &= 6 \\ 2y - z &= 1 \\ x + 4y + 2z &= 7 \end{aligned}$ (b) $\begin{aligned} x + 2y + 3z &= 2 \\ 2y - z &= -3 \\ x + 4y + 2z &= -5 \end{aligned}$ (c) $\begin{aligned} x + 2y + 3z &= -8 \\ 2y - z &= 10 \\ x + 4y + z &= 6 \end{aligned}$

7. (a) $\begin{aligned} x + 2y + z &= 3 \\ -2x - z &= 4 \\ -x + 2y &= 7 \end{aligned}$ (b) $\begin{aligned} x + 2y + z &= 3 \\ -2x - z &= 4 \\ -x + 2y &= 8 \end{aligned}$ (c) $\begin{aligned} x + 2y + z &= 2 \\ -2x - 3y - z &= -3 \\ -x + 2y &= -4 \end{aligned}$

1.2 Matrices and Vectors

Definitions

- An m **x** n **matrix** is a rectangular array of numbers with m rows and n columns.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} = [a_{ij}]$$

- The **dimension** of A is m by n, denoted m x n.
- Two matrices are **equivalent** if they have the same dimension with equal corresponding terms.
- The matrix of all zeros is called the **zero matrix** denoted **0**. For example

$$\mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- The n x n matrix with ones along the diagonal and zeros elsewhere is called the **identity matrix** and is denoted I_n. For example,

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- If $A = [a_{ij}]$, the **transpose** of A, denoted A^T, is defined by $a_{ij}^T = a_{ji}$. (switch rows and columns).

$$\text{If} \quad A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{then} \quad A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}.$$

  If $A^T = A$ then A is called symmetric.

- Diagonal and Triangular Matrices:

Diagonal
$d_{ij} = 0$ if $i \neq j$

$$\begin{bmatrix} d_{11} & 0 & 0 & 0 \\ 0 & d_{22} & 0 & 0 \\ 0 & 0 & d_{33} & 0 \\ 0 & 0 & 0 & d_{44} \end{bmatrix}$$

Upper Triangular
$u_{ij} = 0$ if $i > j$

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Lower Triangular
$l_{ij} = 0$ if $i < j$

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}$$

Matrix Addition and Scalar Multiplication

- **Matrix Addition:** Matrices of the same dimensions are added term by term.

$$\text{if} \quad A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 6 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{then} \quad A + B = \begin{bmatrix} 3 & 0 & 1 \\ 1 & 1 & 5 \end{bmatrix}$$

- **Scalar Multiplication:** When a matrix is multiplied by a number (scalar) this is called scalar multiplication. Each term in the matrix is multiplied by the scalar.

$$\text{if} \quad A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 6 \end{bmatrix} \quad \text{then} \quad 3A = \begin{bmatrix} 3 & -3 & 3 \\ 0 & 3 & 18 \end{bmatrix}$$

Properties of Matrix Addition and Scalar Multiplication

Assume A and B are matrices of the same dimension and r is a scalar (number).

- Commutative Property of Addition: $A + B = B + A$
- Associative Property of Addition: $(A + B) + C = A + (B + C)$
- Identity property of Addition: $A + \mathbf{0} = A$
- Distributive property of scalar multiplication: $r(A + B) = rA + rB$

Row and Column Vectors

A **row vector** is a 1 by n matrix:

$x = [x_1, x_2, \ldots, x_n]$

A **column vector** is an m by 1 matrix:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

You can get a column vector from a row vector by taking the transpose. For example,

$$[1, 2, 3]^T = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

This notation is regularly used to save space.

The Dot Product of Two Vectors

Two vectors of the same length may be multiplied by the dot product:

$$x \cdot y = [x_1, x_2, \ldots, x_n] \cdot [y_1, y_2, \ldots, y_n] = x_1y_1 + x_2y_2 + \ldots x_ny_n \tag{1.2}$$

- **Example:** Find $x \cdot y$ for $x = [2, 3, 4]$ and $y = [-1, 7, -3]$

  Answer: $x \cdot y = (2)(-1) + (3)(7) + (4)(-3) = -2 + 21 - 12 = 7$

Matrix Multiplication

Matrix multiplication is far more complicated than you might think.
If A is an m x n matrix and B is an n by p matrix, then $AB = C$ is an m x p matrix where

$$C_{ij} = \text{i'th row of A dotted with the j'th column of B} \tag{1.3}$$

- **Example:** If $A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 6 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 2 & 3 \end{bmatrix}$, find AB and BA

  Answers:
  Since A is a 2x3 matrix and B is a 3x2 matrix then AB will be the 2x2 matrix given by

$$\begin{aligned} AB &= \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 6 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 2 & 3 \end{bmatrix} \\ &= \begin{bmatrix} (1)(1) + (-1)(0) + (1)(2) & (1)(-1) + (-1)(1) + (1)(3) \\ (0)(1) + (1)(0) + (6)(2) & (0)(-1) + (1)(1) + (6)(3) \end{bmatrix} \\ &= \begin{bmatrix} 1 + 0 + 2 & -1 - 1 + 3 \\ 0 + 0 + 12 & 0 + 1 + 18 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 12 & 19 \end{bmatrix}. \end{aligned}$$

  Since B is a 3x2 matrix and A is a 2x3 matrix then BA will be the 3x3 matrix given by

$$\begin{aligned} BA &= \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 6 \end{bmatrix} \\ &= \begin{bmatrix} (1)(1) + (-1)(0) & (1)(-1) + (-1)(1) & (1)(1) + (-1)(6) \\ (0)(1) + (1)(0) & (0)(-1) + (1)(1) & (0)(1) + (1)(6) \\ (2)(1) + (3)(0) & (2)(-1) + (3)(1) & (2)(1) + (3)(6) \end{bmatrix} \\ &= \begin{bmatrix} 1 + 0 & -1 - 1 & 1 - 6 \\ 0 + 0 & 0 + 1 & 0 + 6 \\ 2 + 0 & -2 + 3 & 2 + 18 \end{bmatrix} = \begin{bmatrix} 1 & -2 & -5 \\ 0 & 1 & 6 \\ 2 & 1 & 20 \end{bmatrix}. \end{aligned}$$

Notes on the Dot Product

- The dot product requires both vectors to be the same length.
- Technically speaking, the dot product doesn't care if one vector is a row vector and the other is a column vector, as long as they have the same number of terms.
- The result of a dot product is a scalar (number). Because of this, the dot product is often called the scalar product.
- **Properties of the Dot Product**
  Assume x, y, and z are vectors of the same length and r is a scalar (number).
  - $x \cdot y = y \cdot x$
  - $(rx) \cdot y = x \cdot (ry)$
  - $x \cdot (y + z) = x \cdot y + x \cdot z$

Notes on Matrix Multiplication

- In order to calculate AB, the number of columns in A **must** be the same as the number of rows in B. If this is not true then AB is undefined.
- If A is an m x n matrix and B is an n x p matrix, then AB is an m x p matrix.
- **Properties of Matrix Multiplication**
  Assume A, B, and C are of suitable dimensions for the following operations.
  - Associative: $A(BC) = (AB)C$
  - Distributive: $A(B + C) = AB + AC$ and $(A + B)C = AC + BC$
  - Identity: If A is n x n then $AI_n = I_nA = A$.
  - Zero: $A\,\mathbf{0} = \mathbf{0}$ and $\mathbf{0}A = \mathbf{0}$ where $\mathbf{0}$ is the zero matrix of appropriate dimension.

Warning: Order Matters!

1. No Commutative Law! $AB \neq BA$ unless you get very lucky.
   Worse: AB and BA might not have the same dimensions.
   Even worse: AB might be defined while BA is undefined.
2. By definition, the dot product of two vectors always yields a scalar ($x \cdot y = y \cdot x$). However, the dot product can be obtained by matrix multiplication if done correctly. If done incorrectly, it fails miserably.
   Let $x = [1, 2]$ and $y = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, then $x \cdot y = y \cdot x = 11$ and $xy = 11$ but $yx = \begin{bmatrix} 3 & 6 \\ 4 & 8 \end{bmatrix}$.
   In most cases you won't want this last result. To reiterate: **Order Matters!**

Linear Systems and Matrix Equations

Any system of linear equations can be expressed as a matrix equation.

- **Example 1:** The system of equations defined by

$$\begin{array}{ccccccccc} x_1 & - & x_2 & + & x_3 & + & 2x_4 & = & 1 \\ & & x_2 & + & 6x_3 & + & 2x_4 & = & 0 \\ & & x_1 & + & 7x_3 & + & 5x_4 & = & 3 \end{array}$$

can be expressed as $Ax = b$ where

$$\begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 1 & 6 & 2 \\ 1 & 0 & 7 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$$

$$A \qquad\qquad x \quad = \quad b.$$

- **Example 2:** Going the other way, the matrix equation $Ax = b$,

$$\begin{bmatrix} 1 & 0 & -3 \\ 2 & 2 & 9 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 7 \\ -2 \end{bmatrix}$$

$$A \qquad\qquad x \quad = \quad b.$$

represents the linear system

$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

Solving Linear Systems in MATLAB®

In MATLAB® you can solve the system of equations defined by $Ax = b$ with

$$x = A\backslash b. \tag{1.4}$$

Notice, the backslash command in $A\backslash$b is **not** the division symbol. If there is a unique solution, this works perfectly. If there are infinitely many solutions or no solutions, it gets a little tricky. We'll investigate all three situations in the worksheet.

Using MATLAB® with Vectors: Here is MATLAB® syntax for various tasks involving vectors.

`v = [2 4 6]` or `v = [2, 4, 6]` yields a row vector.

`v = [2;4;6]` or `v = [2 4 6]’` yields a column vector.

`v(2)` yields the second element in v.

`v(2:3)` yields elements 2 through 3 of v.

`v(1) = 0` replaces the first term in `v` with zero. Afterwards, `v = [0 4 6]`.

`v(4) = 5` appends a 5 to `v`. Afterwards, `v = [0 4 6 5]`

`[m,n] = size(v)` yields m = 1 and n = 3 (if v is row vector) or m = 3 and n = 1 (if v is a column vector).

`v = 0:0.5:2` yields `v = [0 0.5 1 1.5 2]`.

The file (`Vectors.m`) from the Chapter 1 program repository performs various vector operations.

```
%% Vectors.m
%% Examples of defining and operating on vectors
clc;  % This clears the console (optional)
x = [2,4,20];   % a row vector
y = [1 2 3];    % a row vector, commas are optional
z = [0; 1; 4];  % a column vector or z = [0 1 4]'

x2 = x(2) % Access the second element in x and call it x2
y(3) = 6   % assign the third element in y to be 6
L = length(x)  % yeilds the length of vector x
sx = size(x)    % yeilds [1,3]
sz = size(z)    % yeilds [3,1]
addem = x+y     % vector addition (term by term)
scalmult = 3*x  % multiply vector x by 3
wierdo = x + 2  % actually adds 2 to each term in x

%% Dot Product: x and y are both row vectors here
xy1 = dot(x,y)  % guaranteed if x and y are the same length
xy2 = x*y'      % also works (' is transpose)
xy3 = x'*y      % yeilds a 3x3 matrix (probably not what you want)
xy4 = x*y       % Doesn't work: "Inner matrix dimensions must agree."

%% Potential errors (uncomment to see error reports)
%x*y     % "Inner matrix dimensions must agree."
%x + z   % "Matrix dimensions must agree."
```

Using MATLAB® for Matrices, Vectors, and Solving Equations

Here is MATLAB® syntax for various tasks involving matrices and solving equations.

`A = [1 2 3; 4 5 6; 7 8 9]` or `A = [1, 2, 3; 4, 5, 6; 7, 8, 9]` creates the matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

`[m,n] = size(A)` yields m = number of rows in A and n = the number of columns in A.

`A(1,2)` yields the element in row 1 and column 2 of A.

`A(:,2)` yields the second column of A.

`A(2,:)` yields the second row of A.

`A(1:2,3:4)` yields rows 1 to 2 and columns 3 to 4 of A.

`A([1 3 2],[1 3])` yields rows 1 3 2 and columns 1 3 of A.

`A + B` yields term by term addition (appropriate dimensions required)

`A * B` yields normal matrix multiplication (appropriate dimensions required)

`A∧2` yields the matrix multiplication A * A

`A.∧2` squares each entry in A (notice the period before the ∧).

`3*A` multiplies each entry in A by 3.

`cos(A)` takes the cosine of each term in A.

`eye(n)` yields the n x n identity matrix.

`zeros(n,m)` yields an n x m zero matrix.

`ones(n,m)` yields an n x m matrix of all ones.

`transpose(A)` yields the transpose of A.

`A'` gives the conjugate transpose (or just transpose if real).

`inv(A)` yields the inverse of A if one exists.

`det(A)` yields the determinant of A.

`x = A\b` produces a solution to Ax = b. (backslash).

The MATLAB® code on the next page performs a variety of such tasks.

The file (`Matrices.m`) from the Chapter 1 program repository performs various matrix operations.

```
%% Matrices.m
%% Examples of defining and operating on matrices.
clc   % This clears the command window (optional)
disp('Welcome to Matrices')      % displays a message
M = [1, 2, 3; 4, 5, 6; 7, 8, 9]  % semicolons separate rows
N = [1 0 0; 0 0 1; 0 1 0]        % commas are optional
[r,c]=size(M)      % r=#rows, c=#columns
M23 = M(2,3)       % gives the element in row 2, column 3
col1 = M(:,1)      % yields column one of M
row2 = M(2,:)      % yields row two of M
sumMN = M + N      % Matrix addition, must have same dimensions
TransM = M'        % Transpose of M
M2 = 2 * M         % scalar multiplication
MN = M*N           % Matrix Multiplication (order matters)
NM = N*M           % does not equal M*N
wierdo = M+2       % actually adds 2 to each term in M
y = [1 2 3]        % a row vector
Ny = N*y'          % Multiply N times y' (y' is a column vector)

%% Solving Linear Systems
A = [1 0 -3; 2 2 9; 0 1 5]
b = [8 7 -2]'    % the ' makes b a column vector
% Solve Ax = b for x using Gaussian Elimination
x = A\b          % notice the backslash (not division /)
checkb = A*x     % checkb should equal b

%% Some other stuff you may need
Z32 = zeros(3,2)  % yields a 3x2 zero matrix
Id33= eye(3,3)    % yields a 3x3 identity matrix

%% Potential Errors (uncomment to see reports)
% M*y   %"Inner matrix dimensions must agree."
% M\y   %"Matrix dimensions must agree"
```

Chapter 1.2 Worksheet

1. **(MATLAB®)** Recall the following systems that you solved by hand in Section 1.1.

(a)
$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

(b)
$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

(c)
$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 10 \\ x_1 + 5x_2 &= 6 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

You should have found that (a) had a unique solution $[x_1, x_2, x_3] = [5, 3, -1]$, (b) (inconsistent) had no solutions, and (c) (dependent) had infinitely many solutions of the form $[x_1, x_2, x_3] = [6 - 5t, 2 - \frac{t}{2}, t]$.

Assignment: Use MATLAB® to solve these three systems of equations. You should run into problems. You will get warnings for parts (b) and (c). In part (b) MATLAB® actually gives you a solution that is not a solution and in part (c) MATLAB® does not return a solution.

Chapter 1.2 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1.* Perform the following operations by hand and confirm with **MATLAB®** .

$$A = \begin{bmatrix} 2 & -5 \\ 0 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 2 \\ -1 & 1 \end{bmatrix}, \quad \text{and} \quad x = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$$

(a) Find A^T and B^T

(b) Find $\frac{1}{2}A$ (scalar multiplication)

(c) Find $A + B$

(d) Find $2(A + B)$ and $2A + 2B$.

(e) Find AB and BA.

(f) Find Ax and Bx.

2. Perform the requested operations on the given vectors by hand and confirm with **MATLAB®** .

$$x = [1, 2, 3] \qquad y = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

(a) Find $x \cdot y$ (the dot product of x and y).

(b) Find xy (matrix multiplication). You should get the same answer as part (a).

(c) Find yx (matrix multiplication). You should get a 3x3 matrix.

* **Notice** that the dot product is just a type of matrix multiplication if you line it up correctly. If you don't line it up properly, you will get something you probably didn't want.

3.* Perform the following operations by hand and confirm with **MATLAB®** .

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 4 & -2 \\ 1 & 0 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 2 & -1 \\ 0 & 1 & 0 \\ -1 & 2 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ -5 & 0 \end{bmatrix}, x = \begin{bmatrix} 2 \\ -3 \\ 0 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$$

(a) Find A^T and C^T

(b) Find AB and BA

(c) Find AC and CA. One of these will be undefined.

(d) Find Ax and Cx. One of these will be undefined.

(e) Find the dot product $x \cdot y$.

(f) Find $3x - 2y$

4. Perform the following operations by hand and confirm with **MATLAB®** .

$$A = \begin{bmatrix} -1 & 3 & -2 \\ 10 & 1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 1 \\ 3 & 1 & 8 \end{bmatrix} \quad C = \begin{bmatrix} -1 & 0 & 3 \\ 2 & 1 & -2 \end{bmatrix} \quad x = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad y = \begin{bmatrix} -6 \\ 10 \\ 1 \end{bmatrix}$$

   (a) Find A^T, B^T, and C^T.
   (b) Find AB and BA.
   (c) Find AC and CA. One of these will be undefined.
   (d) Find Ax and Cx
   (e) Find the dot product $x \cdot y$.
   (f) Find $3x - 2y$

5. Express the following single matrix equations as a system of linear equations. That is, you are given A, x, and b for a matrix equation of the form $Ax = b$ and you are to explicitly write out the system of equations it represents.

   (a)* $Ax = b$ where $A = \begin{bmatrix} 5 & 7 & 1 & 1 & 1 \\ 0 & -2 & 4 & 2 & 0 \\ 4 & 0 & -2 & 0 & 3 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$ and $b = \begin{bmatrix} 5 \\ -1 \\ 3 \end{bmatrix}$

   (b) $Ax = b$ where $A = \begin{bmatrix} 0 & 1 & -4 \\ 2 & -2 & 7 \end{bmatrix} \quad x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ and $b = \begin{bmatrix} 17 \\ -1 \end{bmatrix}$

6. Express the following systems of equations in terms of a single matrix equation. That is, determine the matrix A and the vectors x and b so that the system of equations is expressed equivalently as $Ax = b$.

   (a)* $$\begin{aligned} x - 3z &= -5 \\ 2x + y + 2z &= 7 \\ 3x + 2y + z &= 7 \end{aligned}$$

   (b) $$\begin{aligned} x_1 + x_2 + 2x_3 - x_4 &= 8 \\ -x_1 - 2x_2 + 3x_3 + 2x_4 &= 1 \\ 3x_1 - 4x_3 &= 10 \end{aligned}$$

7.* **(MATLAB®** **)** Solve the following systems using MATLAB® .

   (a) $$\begin{aligned} x - 3z &= -5 \\ 2x + y + 2z &= 7 \\ 3x + 2y + z &= 7 \end{aligned}$$

   (b) $$\begin{aligned} x_1 + x_2 + 2x_3 &= 8 \\ -x_1 - 2x_2 + 3x_3 &= 1 \\ 3x_1 - 7x_2 + 4x_3 &= 10 \end{aligned}$$

8. **(MATLAB®** **)** Solve the following systems using MATLAB® .

   (a) $$\begin{aligned} x + 2y + 3z &= -8 \\ 2y - z &= 10 \\ x + 4y + z &= 6 \end{aligned}$$

   (b) $$\begin{aligned} x_1 + x_2 &= -0.25 \\ -x_1 + 3x_3 &= 5.50 \\ -7x_2 + 4x_3 &= 13.25 \end{aligned}$$

1.3 Inverses and Determinants

If you came across the scalar equation $ax = b$ and you want to solve for x, you would just divide both sides of the equation by a and get $x = b/a$. This is all well and good if $a \neq 0$. If $a \neq 0$ then a has a multiplicative inverse which we denote by $1/a$ or $^{-1}$. If $a = 0$ then we have problems. Specifically, zero has no multiplicative inverse. This is summed up by the rule: *You can't divide by zero.*

It would be nice if there was a simple rule like this for matrix equations in the form $Ax = b$. It turns out, there is. It goes like this: *If the determinant of a matrix is zero, it has no inverse.* This section describes the process of determining whether or not a matrix has an inverse and, if there is one, how to get it.

Definition of the Inverse of a Matrix

Assume A is a square, n x n, matrix. If there exists an n x n matrix B such that

$$AB = BA = I_n$$

then B is called the **inverse** of A, denoted by A^{-1}, and A is called **invertible**. In this case A is also called **nonsingular**. If A does not have an inverse it is called **singular**.

Inverse of a 2 by 2 matrix

$$\text{If} \quad A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{then} \quad A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}. \tag{1.5}$$

Notice: The number $ad - bc$ determines whether A has an inverse or not. This number is called the **determinant** of a 2 by 2 matrix.

Solution of a Matrix Equation

If A is an n by n invertible matrix, then the system of equations expressed by $Ax = b$ has a solution defined by $x = A^{-1}b$.

$$Ax = b \quad \rightarrow \quad x = A^{-1}b \tag{1.6}$$

- **Example:** Use the method above to solve the system $\begin{array}{rcrcl} x_1 & + & 2x_2 & = & 5 \\ 3x_1 & + & 4x_2 & = & 6 \end{array}$

  Answer:

  Here, $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $b = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$ so $A^{-1} = \dfrac{1}{(1)(4) - (2)(3)} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} = \dfrac{-1}{2} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$

  and

  $$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = A^{-1}b = \frac{-1}{2} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \frac{-1}{2} \begin{bmatrix} 8 \\ -9 \end{bmatrix} = \begin{bmatrix} -4 \\ 4.5 \end{bmatrix}$$

  We can check this solution by substituting $x_1 = -4$ and $x_2 = 4.5$ into the original system of equations

  $\begin{array}{rcrclclcl} x_1 & + & 2x_2 & = & -4 + 2(4.5) & = & -4 + 9 & = & 5 \\ 3x_1 & + & 4x_2 & = & 3(-4) + 4(4.5) & = & -12 + 18 & = & 6 \end{array}$ and the solution is valid.

Inverse Theorems: Assume A and B are both invertible matrices and k is a scalar.

(a) $(AB)^{-1} = B^{-1}A^{-1}$

(b) $(A^{-1})^{-1} = A$

(c) $(A^T)^{-1} = (A^{-1})^T$

(d) $(kA)^{-1} = \frac{1}{k}A^{-1}$

Getting the inverse of larger matrices by row operations

You want to find the matrix X such that

$$AX = I_n.$$

Set up the augmented matrix $[A|I_n]$ where I_n is the nxn identity matrix. Perform row operations until A is equivalent to I_n then the right half is the inverse of A. In other words convert $[A|I_n] \sim [I_n|A^{-1}]$ using row operations (if possible).

- **Example:** Find the inverse of $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 9 & 12 \\ 2 & 7 & 5 \end{bmatrix}$.

  Set up the augmented matrix $[A|I_3]$ and perform row operations to get $[I_3|A^{-1}]$

$$[A|I_3] = \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 4 & 9 & 12 & 0 & 1 & 0 \\ 2 & 7 & 5 & 0 & 0 & 1 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 0 & -4 & 1 & 0 \\ 0 & 3 & -1 & -2 & 0 & 1 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 0 & -4 & 1 & 0 \\ 0 & 0 & -1 & 10 & -3 & 1 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 0 & -4 & 1 & 0 \\ 0 & 0 & 1 & -10 & 3 & -1 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 2 & 0 & 31 & -9 & 3 \\ 0 & 1 & 0 & -4 & 1 & 0 \\ 0 & 0 & 1 & -10 & 3 & -1 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 39 & -11 & 3 \\ 0 & 1 & 0 & -4 & 1 & 0 \\ 0 & 0 & 1 & -10 & 3 & -1 \end{array}\right] = [I_3|A^{-1}]$$

Now A^{-1} is the right half of the augmented matrix.

$$A^{-1} = \begin{bmatrix} 39 & -11 & 3 \\ -4 & 1 & 0 \\ -10 & 3 & -1 \end{bmatrix}.$$

The Determinant of a 2 by 2 Matrix

The determinant of a 2 by 2 matrix A, denoted $\det(A)$ or $|A|$, is given by

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc \tag{1.7}$$

Note: Earlier it was shown that if $\det(A) = 0$ then A does not have an inverse. This carries over to larger matrices but we must first define and be able to calculate the determinant of larger matrices.

The Determinant of Larger Matrices

Assume A is a square n by n matrix. We first need a couple definitions before we calculate its determinant.

The ij**th** **minor of** A is the $(n-1)$ x $(n-1)$ matrix M_{ij} resulting from A when the ith row and jth column are removed.

The ij**th** **cofactor of** A is denoted A_{ij} and defined by

$$A_{ij} = (-1)^{i+j}\,|M_{ij}|$$

The determinant of A, denoted $\det(A)$ or $|A|$, is given by

$$\det(A) = a_{11}\,A_{11} + a_{12}\,A_{12} + \ldots + a_{1n}\,A_{1n}. \tag{1.8}$$

In (1.8), A has been expanded along the first row. It can also be expanded along any row i

$$\det(A) = a_{i1}\,A_{i1} + a_{i2}\,A_{i2} + \ldots + a_{in}\,A_{in} \tag{1.9}$$

or any column j

$$\det(A) = a_{1j}\,A_{1j} + a_{2j}\,A_{2j} + \ldots + a_{nj}\,A_{nj} \tag{1.10}$$

to produce the same determinant.

- **Example:** Find the determinant of $A = \begin{bmatrix} 2 & 4 & 6 \\ 0 & 1 & 0 \\ 2 & 3 & -1 \end{bmatrix}$

  - by expanding along the first row

    $\det(A) = (-1)^{(1+1)}(\mathbf{2})\begin{vmatrix} 1 & 0 \\ 3 & -1 \end{vmatrix} + (-1)^{(1+2)}(\mathbf{4})\begin{vmatrix} 0 & 0 \\ 2 & -1 \end{vmatrix} + (-1)^{(1+3)}(\mathbf{6})\begin{vmatrix} 0 & 1 \\ 2 & 3 \end{vmatrix}$

    $\det(A) = (2)(-1-0) - (4)(0-0) + (6)(0-2) = -2 - 0 - 12 = \mathbf{-14}$.

  - by expanding along the second row

    $\det(A) = (-1)^{(2+1)}(\mathbf{0})\begin{vmatrix} 4 & 6 \\ 3 & -1 \end{vmatrix} + (-1)^{(2+2)}(\mathbf{1})\begin{vmatrix} 2 & 6 \\ 2 & -1 \end{vmatrix} + (-1)^{(2+3)}(\mathbf{0})\begin{vmatrix} 2 & 4 \\ 2 & 3 \end{vmatrix}$

    $\det(A) = -(0)(-4-18) + (1)(-2-12) - (0)(6-8) = 0 - 14 - 0 = \mathbf{-14}$.

  - by expanding along the first column

    $\det(A) = (-1)^{(1+1)}(\mathbf{2})\begin{vmatrix} 1 & 0 \\ 3 & -1 \end{vmatrix} + (-1)^{(2+1)}(\mathbf{0})\begin{vmatrix} 4 & 6 \\ 3 & -1 \end{vmatrix} + (-1)^{(3+1)}(\mathbf{2})\begin{vmatrix} 4 & 6 \\ 1 & 0 \end{vmatrix}$

    $\det(A) = (2)(-1-0) - (0)(-4-18) + (2)(0-6) = -2 - 0 - 12 = \mathbf{-14}$.

Warning: Using the cofactor expansion to calculate the determinant of a matrix can be impossibly time consuming even for a computer when the matrix gets too big. There is a way around this. We will not go into the procedure but it reduces to performing Gaussian elimination while keeping track of the row operations. It is also worth noting that problems can arise even when the determinant is not zero.

Determinant Theorems:

(a) If A is a triangular matrix (upper or lower) then $\det(A) = a_{11}\, a_{22} \ldots a_{nn}$

(b) $\det(A^T) = \det(A)$

(c) $\det(AB) = \det(A)\det(B)$

(d) If A is invertible then $\det(A^{-1}) = \dfrac{1}{\det(A)}$.

Super Theorem - Version 1: Let A be an $n\ x\ n$ matrix. The following are equivalent (TFAE)

- $Ax = b$ has a unique solution for any b.
- A is invertible (nonsingular or has an inverse).
- $A \sim I_n$. (A is row equivalent to the identity matrix).
- $\det(A) \neq 0$.

Using MATLAB® to get Inverses and Determinants

```
%% This is Inverses_and_Determinants.m

A = [2 4 6; 0 1 0; 2 3 -1]
det(A)  % The determinant is -14.
inv(A)  % It has an inverse.

B = [2 3 1; 0 1 0; 2 4 1];
det(B)  % The determinant is 0.
inv(B)  % You get a warning.

%% INVERSE THEOREM (b)
inv(inv(A)) % The inverse of the inverse of A is A

%% DETERMINANT THEOREM (a)
D = [2 3 1;
     0 3 2;
     0 0 4] % upper-triangular matrix
detD = det(D)  % determinant = 2 * 3 * 4 = 24
```

MATLAB® Syntax:

`det(A)` produces $\det(A)$.

`inv(A)` produces A^{-1}.

If there is no inverse, MATLAB® generates an error message.

More Matrix Equations

Earlier we solved equations of the form $Ax = b$ using Gaussian Elimination. We did this by hand and with MATLAB® using the backslash command `x = A\b`. We then saw how you can get this solution symbolically using matrix inverses by $x = A^{-1}b$. Here we look at more complicated matrix equations.

- **Examples:** Assume A and B are 3 x 3 invertible matrices and x, y, and b are 3 x 1 column vectors.

  1. Solve $Ax + y = b$ for x.

  $$\begin{aligned} Ax + y &= b \\ Ax &= b - y \\ A^{-1}Ax &= A^{-1}(b - y) \\ I_3 x &= A^{-1}(b - y) \\ x &= A^{-1}(b - y). \end{aligned}$$

  This solution is certain to exist because A was assumed to be invertible.

  2. Solve $A(x + y) = b$ for x.

  $$\begin{aligned} A(x + y) &= b \\ x + y &= A^{-1}b \\ x &= A^{-1}b - y. \end{aligned}$$

  This solution is certain to exist because A was assumed to be invertible.

  3. Solve $A(Bx + y) = b$ for x.

  $$\begin{aligned} A(Bx + y) &= b \\ Bx + y &= A^{-1}b \\ Bx &= A^{-1}b - y \\ x &= B^{-1}(A^{-1}b - y). \end{aligned}$$

  This solution is certain to exist because A and B were assumed to be invertible.

  4. Solve $A(x + y) = Bx + b$ for x.

  $$\begin{aligned} A(x + y) &= Bx + b \\ Ax + Ay &= Bx + b \\ Ax - Bx &= -Ay + b \\ (A - B)x &= -Ay + b. \end{aligned}$$

  At this point you may want to claim the solution is

  $$x = (A - B)^{-1}\left[-Ay + b\right].$$

  However, this may not exist because there is no guarantee that $A - B$ has an inverse. So we can't be sure whether there is a unique solution until we get the determinant of $A - B$.

Chapter 1.3 Worksheet

1. **(MATLAB®)** Recall the systems you solved by hand in Section 1.1 and with MATLAB® in 1.2:

(a)
$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

(b)
$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

(c)
$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 10 \\ x_1 + 5x_2 &= 6 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

You should have found that (a) had a unique solution $[x_1, x_2, x_3] = [5, 3, -1]$, (b) (inconsistent) had no solutions, and (c) (dependent) had infinitely many solutions of the form $[x_1, x_2, x_3] = [6 - 5t, 2 - \frac{t}{2}, t]$.

Assignment: Enter the coefficient matrices into MATLAB® , then find the determinants and inverses of these three matrices. You should run into problems.

For part (a), you should get a non-zero determinant and nice inverse matrix. For part (b), you should get something very close to zero for the determinant, and when you try to find its inverse you should get a warning and an inverse with crazy huge numbers. For part (c) you should get zero for the determinant and when you try to get its inverse you get a message that the matrix is singular and it returns a matrix of non-numbers.

Chapter 1.3 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. Find the determinants by hand. If the determinant is non-zero, find the inverse of the matrix again by hand. Check your answer with MATLAB® .

   (a)* $A = \begin{bmatrix} 1 & 2 \\ 4 & 7 \end{bmatrix}$

   (b) $B = \begin{bmatrix} -3 & 1 \\ -4 & -2 \end{bmatrix}$

   (c)* $C = \begin{bmatrix} 1 & 0 & -2 \\ -3 & 1 & 4 \\ 2 & -3 & 4 \end{bmatrix}$

   (d) $D = \begin{bmatrix} 1 & -2 & 1 \\ 4 & -7 & 3 \\ -2 & 6 & -4 \end{bmatrix}$

2. Find the value(s) of a for which the given matrix A does **NOT** have an inverse.

   (a) $A = \begin{bmatrix} a & 5 \\ 1 & 3 \end{bmatrix}$. There is one solution.

   (b)* $A = \begin{bmatrix} (1-a) & -2 \\ 1 & (4-a) \end{bmatrix}$. There are two solutions.

   (c) $A = \begin{bmatrix} (2-a) & -2 \\ 1 & (-1-a) \end{bmatrix}$. There are two solutions.

   (d)* $A = \begin{bmatrix} 1 & a & 2 \\ 4 & -1 & 3 \\ -2 & 4 & -1 \end{bmatrix}$. There is one solution.

   (e) $A = \begin{bmatrix} 1 & 0 & a \\ -3 & 1 & 4 \\ 2 & -3 & 4 \end{bmatrix}$. There is one solution.

3.* **(MATLAB®)** Consider the matrix A, the three b, and matrix B given below.

$$A = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 4 & -3 & 8 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad b_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Notice, the columns of B are the vectors b_1, b_2, and b_3.

(a) Use MATLAB® to find A^{-1} and use this to solve the three equations

$$Ax_1 = b_1 \qquad Ax_2 = b_2 \qquad Ax_3 = b_3$$

(b) Now use the MATLAB® backslash command (`x = A\b`) to solve the same equations. Do your answers agree?

(c) Now, let $X = A^{-1}B$. What do you notice about the matrix X?

4.* **(MATLAB®)** In the MATLAB® file `InversesAndDeterminants.m`, Inverse theorem (b) and determinant theorem (a) were demonstrated numerically. Write a MATLAB® program that demonstrates inverse theorems (a), (c), (d), and the determinant theorems (b), (c) and (d).

5. **(MATLAB®)** Given the following vectors and matrices, solve the requested matrix equation for x. You should first simplify the answer using matrix algebra and then find the solution using MATLAB®. If there is not a unique solution, state how you know.

$$A = \begin{bmatrix} 1 & 0 & -2 \\ -3 & 1 & 4 \\ 2 & -3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad y = \begin{bmatrix} 3 \\ -13 \\ 20 \end{bmatrix} \quad b = \begin{bmatrix} -5 \\ 7 \\ 20 \end{bmatrix}$$

(a)* $Ax + y = b$

(b) $A(x + y) = b$

(c)* $A(Bx + y) = b$

(d) $A(x + y) = Bx + b$

(e)* $ABx = 2Ax + b$ (You should run into problems.)

(f) $Ax = A(By + b)$

(g)* $ABx = BAy$

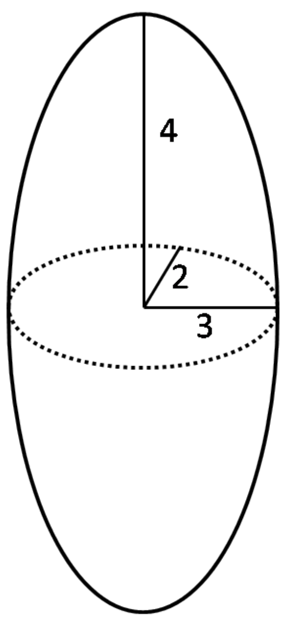
(h) $A(y - 2x) = 4b - Ay$

1.4 Vector Spaces: $\mathbb{R}^2$ and $\mathbb{R}^3$

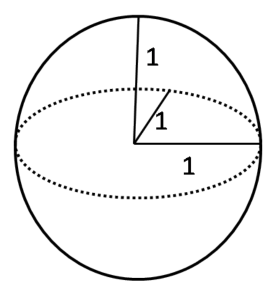
Motivation

When modeling three dimensional interactions we want to know where an object is and if it has hit anything else as it moves around. The object might have a complex shape, but we can enclose it in an ellipsoid (example below) and then determine if the ellipsoid has hit anything. This is much easier but not as easy as it gets because even an ellipsoid is not uniform in shape. It is extra difficult when trying to determine a collision response. The **Unit Sphere** is really nice. If the center gets less than one unit from another object - there is a collision. Furthermore, the response is easier to predict. Ask anyone who tries to grab a football after it hits the ground. It would be a lot easier if it was spherical. So, if we can work in a world where all ellipsoids are actually spheres, life just got a lot easier. Making the transformation from the real world with ellipsoids, to a convenient world of spheres, requires a *change of basis* transformation. After this transformation is made, we do all of our calculations in the convenient world of spheres and then transform back to the world of ellipsoids to see what actually happens. That's the goal, but there is a lot to do first.

Ellipsoid
radius vector $= [r_x, r_y, r_z] = [3, 2, 4]$



Unit Sphere
radius vector $= [r_x, r_y, r_z] = [1, 1, 1]$



The different *worlds* of ellipses and spheres are actually different mathematical representations of the same space - a vector space. In this chapter, we will not be going into all the different kinds of vector spaces. We will cover the material needed to handle basic tasks in 2 and 3 dimensional space. We will start with the general properties of any vector space and investigate the two most important vector spaces for our purposes. These are $\mathbb{R}^2$ and $\mathbb{R}^3$.

General Vector Spaces

A vector space is a set V, whose elements are called vectors. If a vector x is in the vector space, we say $x \in V$. A vector space has two operations; addition and scalar multiplication. These operations must satisfy the following properties.

1. Addition is a binary operation on V which is both commutative and associative.
   (a) If $x \in V$ and $y \in V$ then $x + y \in V$. **closure under addition**
   (b) For all x, y, and z in V, $(x + y) + z = x + (y + z)$. **associative property of addition**
   (c) If x and y are in V, then $x + y = y + x$. **commutative property of addition**
2. There is a vector $\mathbf{0} \in V$ such that $x + \mathbf{0} = x$ for all $x \in V$. **additive identity**
3. If $x \in V$ then there is a vector $-x$ such that $x + (-x) = \mathbf{0}$. **additive inverse**
4. If $x \in V$ and α is a scalar (real number), then $\alpha x \in V$. **closure under scalar multiplication**
5. Scalar Multiplication Properties.
   (a) If x and y are in V and α is a scalar, then $\alpha(x + y) = \alpha x + \alpha y$. **distributive property 1**
   (b) If $x \in V$ and α and β are scalars, then $(\alpha + \beta)x = \alpha x + \beta x$. **distributive property 2**
   (c) If $x \in V$ and α and β are scalars, then $\alpha(\beta x) = (\alpha\beta)x$. **associative law of scalar mult.**
   (d) For every vector $x \in V$, $1\,x = x$. **scalar multiplicative identity**

The Vector Space $\mathbb{R}^3$: This represents the three dimensional space in which we live and may try to recreate on the computer. In this notation, the funny looking $\mathbb{R}$ represents real numbers, and the three represents how many of them we have. A vector in $\mathbb{R}^3$ is essentially a point in 3-space. For example the vector $[2, -3, 4]^T$ is the point obtained by going 2 units in the x-direction, -3 units in the y-direction, and 4 units in the z-direction.

$$\mathbb{R}^3 = \left\{ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \text{ where } x_1, x_2, x_3 \in \mathbb{R} \right\} \qquad \textbf{example: } x = \begin{bmatrix} 2 \\ -3 \\ 4 \end{bmatrix} \in \mathbb{R}^3$$

The Vector Space $\mathbb{R}^2$: This represents the 2 dimensional space in the plane. It is actually a vector subspace of $\mathbb{R}^3$. A vector in $\mathbb{R}^2$ is essentially a point in 2-space. For example the vector $[2, -2]$ is the point obtained by going 2 units in the x-direction and -2 units in the y-direction.

$$\mathbb{R}^2 = \left\{ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \text{ where } x_1, x_2 \in \mathbb{R} \right\} \qquad \textbf{example: } x = \begin{bmatrix} 2 \\ -2 \end{bmatrix} \in \mathbb{R}^2$$

The Vector Space $\mathbb{R}^n$: This vector space looks just like $\mathbb{R}^2$ or $\mathbb{R}^3$ except each vector has n terms. While it would takes some time to prove, you should be able to convince yourself that $\mathbb{R}^n$ is a vector space. In this space, a scalar (α) is just a real number.

Linear Combinations

Let $v_1,\ v_2,\ \ldots,\ v_n$ be vectors in a vector space V, and $a_1,\ a_2,\ \ldots,\ a_n$ be scalars from the same vector space. Any vector x in the form

$$x = a_1v_1 + a_2v_2 + \ldots + a_nv_n \tag{1.11}$$

is called a **linear combination** of $v_1,\ v_2,\ \ldots,\ v_n$. The scalars $a_1,\ a_2,\ \ldots,\ a_n$, are called the **coefficients** of the linear combination.

Generating Linear Combinations

In practice, linear combinations are generated via matrix multiplication by first creating a column vector of the coefficients $a = [a_1, a_2, \ldots a_n]^T$. Then multiply this on the left by the matrix M whose columns are $v_1 \ldots v_n$.

$$x = a_1v_1 + a_2v_2 + \ldots + a_nv_n = Ma = \left[\begin{array}{c|c|c|c} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \ldots & v_n \\ \vdots & \vdots & \vdots & \vdots \end{array}\right] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \tag{1.12}$$

- **Example in $\mathbb{R}^2$:** Let $v_1 = [-1, 0]$ and $v_2 = [3, 4]$ find the linear combination $x = -2v_1 + 5v_2$.

  Answer: First we define our matrix M whose columns are v_1 and v_2 then

$$x = Ma = \begin{bmatrix} -1 & 3 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 5 \end{bmatrix} = \begin{bmatrix} 17 \\ 20 \end{bmatrix}$$

Determining Linear Combinations

Often we will want to know what linear combination of a given set of vectors produces a specific vector. In this case we are looking for the coefficients of the linear combination. This task is accomplished by solving a linear system.

- **Example in $\mathbb{R}^3$:** Consider the vectors: $v_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$, $v_2 = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$, $v_3 = \begin{bmatrix} 2 \\ -1 \\ 6 \end{bmatrix}$.

  Express the vector $x = [4, -1, 14]^T$ as a linear combination of $v_1,\ v_2,\ v_3$. That is, find $a_1,\ a_2,\ a_3$ such that $x = a_1v_1 + a_2v_2 + a_3v_3$.

  Answer: You must solve the matrix equation $V\mathbf{a} = x$ for $\mathbf{a}$, or

$$\begin{bmatrix} 1 & 1 & 2 \\ -1 & 0 & -1 \\ 2 & 3 & 6 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 14 \end{bmatrix}$$

  You can solve this by hand or use software to obtain the solutions $[a_1, a_2, a_3] = [-2, 0, 3]$.

<u>The Standard Bases for $\mathbb{R}^2$ and $\mathbb{R}^3$</u>

In a general vector space, all vectors must be identifiable as a unique **linear combination** of the vector space's **basis** vectors. Each vector is then uniquely identified by the coefficients used in this linear combination. The **standard basis** is really nice because the coefficients in this linear combination are just the numbers in the vector. This seems redundant and it is. However, when we start using different bases, it is not.

- The **standard basis** for $\mathbb{R}^2$ consists of the two vectors e_1 and e_2 defined by

$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

This basis works wonderfully well because any vector in $\mathbb{R}^2$ can be easily expressed as a **linear combination** of these 2 vectors. For example, the vector $[2, -2]^T$ can be expressed as

$$\begin{bmatrix} 2 \\ -2 \end{bmatrix} = 2\begin{bmatrix} 1 \\ 0 \end{bmatrix} - 2\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2e_1 - 2e_2.$$

It's that simple.

- The **standard basis** for $\mathbb{R}^3$ consists of the three vectors e_1, e_2, and e_3 defined by

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Again, this basis works wonderfully well because any vector in $\mathbb{R}^3$ can be easily expressed as a **linear combination** of these 3 vectors. For example, the vector $[2, -3, 4]^T$ can be expressed as

$$\begin{bmatrix} 2 \\ -3 \\ 4 \end{bmatrix} = 2\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - 3\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 4\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 2e_1 - 3e_2 + 4e_3$$

Chapter 1.4 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1.* Verify that the linear combination of vectors on the left side of the = sign can be expressed as the matrix multiplication on the right.

$$a_1 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} + a_2 \begin{bmatrix} -2 \\ 1 \\ -3 \end{bmatrix} + a_3 \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 \\ 0 & 1 & 5 \\ 2 & -3 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

2. Consider the vectors $v_1 = [1, 4]$ and $v_2 = [2, 7]$. Express the following vectors as a linear combination of these vectors. That is, find a_1 and a_2 such that $x = a_1v_1 + a_2v_2$. You should set this up by hand but are welcome to use MATLAB® to find the coefficients.

   (a)* $x = [-2, 3]$

   (b) $x = [0, -4]$

3. Consider the vectors $v_1 = [1, -3, 2]$, $v_2 = [0, 1, -3]$, and $v_3 = [-2, 4, 4]$. Express the following vectors as a linear combination of these vectors. That is, find a_1, a_2, and a_3 such that $x = a_1v_1 + a_2v_2 + a_3v_3$. You should set this up by hand but are welcome to use MATLAB® to find the coefficients.

   (a)* $x = [1, 1, 1]$

   (b) $x = [7, -3, 0]$

4. Express the vector $[1, 7, 8]^T$ as a linear combination of the vectors $[2, 0, 0]^T$, $[0, \frac{1}{2}, 0]^T$, $[0, 0, 4]^T$.

1.5 Linear Independence, Span, Basis, Dimension

In the previous section we saw the two primary vector spaces that we will use. These were $\mathbb{R}^2$ and $\mathbb{R}^3$. We also saw the convenient standard basis for each of these vector spaces. However, every vector space has infinitely many bases and using an alternative basis will prove to be helpful down the road. Here we will investigate what makes a basis and what a basis tells us about the vector space.

Refresher

Here is some material from the last section that will be important here.

- **Linear Combinations:** Let $v_1,\ v_2,\ \ldots,\ v_n$ be vectors in a vector space V, and $a_1,\ a_2,\ \ldots,\ a_n$ be scalars from the same vector space. Any vector x in the form

$$x = a_1\ v_1 + a_2\ v_2 + \ldots + a_n\ v_n$$

  is called a **linear combination** of $v_1,\ v_2,\ \ldots,\ v_n$. The scalars $a_1,\ a_2,\ \ldots,\ a_n$, are called the **coefficients** of the linear combination. A linear combination of vectors can be obtained with matrix multiplication.

$$x = a_1\ v_1 + a_2\ v_2 + \ldots + a_n\ v_n = Ma = \left[\begin{array}{c|c|c|c} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \ldots & v_n \\ \vdots & \vdots & \vdots & \vdots \end{array}\right] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

- The **standard basis** for $\mathbb{R}^2$ consists of the two vectors e_1 and e_2 defined by

$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

- The **standard basis** for $\mathbb{R}^3$ consists of the three vectors e_1, e_2, and e_3 defined by

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Preliminary Definition of a Basis: A basis for a given vector space is any smallest collection of vectors from that space that can be linearly combined to obtain any other vector in that space. Or, put in a way containing definitions to follow: A basis is any collection of **linearly independent vectors** from the vector space that **span** the entire vector space.

It seems complicated. It's not so bad. Here we go.

Linear Independence

Here we define what it means for a set of vectors to be linearly independent. Two equivalent definitions are given. The first makes more intuitive sense and the second is better for actually determining whether or not a set of vectors are linearly independent. Each definition starts with a set of vectors $\{v_1, v_2, \ldots, v_n\}$ in a vector space V.

Linear Independence - Intuitive Definition: The vectors are said to be linearly dependent if at least one of them can be obtained as a linear combination of the others. That is, at least one of the vectors, say v_k, can be obtained by

$$a_1\, v_1 + \ldots + a_n\, v_n = v_k, \tag{1.13}$$

where v_k is not included on the left hand side of this equation. If none of the vectors can be expressed as a linear combination of the others, the vectors are said to be linearly independent.

Linear Independence - Formal Definition: The vectors are said to be linearly dependent if the zero vector can be obtained as a non-zero linear combination of these vectors. That is,

$$a_1\, v_1 + a_2\, v_2 + \ldots + a_n\, v_n = \mathbf{0} \tag{1.14}$$

where at least one of the a_k's is not zero. If the vectors are not linearly dependent they are said to be linearly independent.

♣ Mini-Proof

This mini-proof outlines why the intuitive and formal definitions of linear independence are equivalent. Suppose $v_1, v_2, \ldots, v_n$ are linearly dependent vectors.

Intuitive Definition ⇒ Formal Definition

By equation (1.13) from the intuitive definition, there exists v_k such that

$$\begin{aligned} a_1\, v_1 + \ldots + a_n\, v_n &= v_k \\ a_1 v_1 + \ldots - 1\, v_k + \ldots + a_n v_n &= \mathbf{0} \end{aligned}$$

Since the coefficient of v_k is not zero, the vectors are linearly dependent by equation (1.14) from the formal definition.

Formal Definition ⇒ Intuitive Definition

By equation (1.14) from the formal definition, there exists $a_k \neq 0$ such that

$$\begin{aligned} a_1\, v_1 + \ldots + a_k\, v_k + \ldots + a_n v_n &= \mathbf{0} \\ a_1\, v_1 + \ldots + a_n\, v_n &= -a_k\, v_k \\ -a_1/a_k\, v_1 - \ \ldots\ - a_n/a_k\, v_n &= v_k \end{aligned}$$

Since v_k is a linear combination of the other vectors, the vectors are linearly dependent by equation (1.13) from the intuitive definition. ♣

While these two definitions of linear independence are equivalent, determining linear independence by the intuitive definition is difficult unless it is obvious that one of the vectors is a linear combination of the others. Usually this is obvious only if one of the vectors happens to be a scalar multiple of one of the others. When this is not the case, the formal definition of linear independence is much easier to use.

Determining Linear Independence in $\mathbb{R}^n$

Here we determine whether or not a set of n vectors are linearly independent in the vector space $\mathbb{R}^n$. This is done by setting up equation (1.14) in the matrix form $Ma = \mathbf{0}$.

$$a_1\, v_1 + a_2\, v_2 + \ldots + a_n\, v_n = \mathbf{0} \quad \Longleftrightarrow \quad Ma = \left[\begin{array}{c|c|c|c} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \ldots & v_n \\ \vdots & \vdots & \vdots & \vdots \end{array}\right] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

If the system has a non-zero solution (at least one term in a is not zero) then the vectors are dependent. If the only solution is $a_1 = a_2 = \ldots = a_n = 0$, then the vectors are linearly independent.

- **Example in $\mathbb{R}^2$:** Verify that the vectors $v_1 = [-1, 0]$ and $v_2 = [3, 4]$ are linearly independent.

  Answer: We must ask ourselves: Is there a non-zero solution to the following equation?

  $$Ma = \begin{bmatrix} -1 & 3 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

  Since det(M) = -4, Super Theorem - Version 1 states that M has an inverse and $a = M^{-1}\mathbf{0} = [0, 0]^T$. Since the only solution is the zero vector, we have proven that $a_1 = a_2 = 0$ and therefore v_1 and v_2 are linearly independent. In this particular case there were only two vectors. We could have used the intuitive definition of linear independence and observed that the vectors are not scalar multiples of each other and must be independent. This shortcut can only be used if there are only two vectors.

Span: The vectors $v_1,\ v_2,\ \ldots,\ v_n$ in a vector space V, are said to span V, if every vector in V can be written as a linear combination of them. That is, for every $x \in V$, there are scalars, $a_1,\ a_2,\ \ldots,\ a_n$, such that $x = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n$

In Practice for $\mathbb{R}^n$:
Create the matrix M whose columns are $v_1 \ldots v_n$ and determine if the system $Ma = x$ has a solution for any $x \in V$.

$$Ma = \left[\begin{array}{c|c|c|c} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \ldots & v_n \\ \vdots & \vdots & \vdots & \vdots \end{array}\right] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- **Example in $\mathbb{R}^2$:** Verify that the vectors $v_1 = [-1, 0]$ and $v_2 = [3, 4]$ span $\mathbb{R}^2$.

  Answer: Let $x = [x_1, x_2]^T$ be any vector in $\mathbb{R}^2$. Create the matrix M whose columns are v_1 and v_2. Can we solve the equation $Ma = x$ for a?

  $$Ma = \begin{bmatrix} -1 & 3 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

  Yes! Since the det(M) = -4, Super Theorem - Version 1 states that M has an inverse and $a = M^{-1}x$. So, we have proven that v_1 and v_2 span $\mathbb{R}^2$.

<u>**Basis:**</u> A finite set of vectors $\{v_1,\ v_2,\ \ldots,\ v_n\}$ is a **basis** for a vector space V if and only if

1. $\{v_1,\ v_2,\ \ldots,\ v_n\}$ are linearly independent and
2. $\{v_1,\ v_2,\ \ldots,\ v_n\}$ spans V.

In Practice for $\mathbb{R}^n$: We create the n x n matrix M whose columns are $v_1 \ldots v_n$. If $\det(M) = 0$, the vectors do not form a basis, if $\det(M) \neq 0$, the vectors do form a basis.

- **Example in $\mathbb{R}^2$:** Verify that the vectors $v_1 = [-1, 0]$ and $v_2 = [3, 4]$ form a basis for $\mathbb{R}^2$.

  Answer: On the previous page we demonstrated that v_1 and v_2 were linearly independent and span $\mathbb{R}^2$ which proves they form a basis. A quick way to all three of these conclusions is observing that the determinant of M was nonzero.

- **Example in $\mathbb{R}^3$:** Prove the vectors $v_1 = [3, 0, 0]$, $v_2 = [0, 2, 0]$, $v_3 = [0, 0, 4]$ form a basis for $\mathbb{R}^3$.

  Answer: Set up the matrix with v_1, v_2, and v_3 as the columns and check the determinant.

$$\det(M) = \det\left(\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}\right) = 24$$

  Since the determinant of this matrix is not zero, we have a basis.
  Note: If we have an ellipsoid with radius vector [3,2,4], this radius vector becomes [1,1,1] with respect to this basis. The ellipsoid becomes a unit sphere!

<u>**Dimension:**</u> The dimension of a vector space V is the number of vectors in every basis for V. Note: Once you find a basis for V you know that the dimension of V is the number of vectors in that basis and all bases will have the same number of vectors = the dimension of V.

- **Example in $\mathbb{R}^2$:** Verify that the vector space $\mathbb{R}^2$ has dimension 2.

  Answer: Above, we determined that the two vectors $v_1 = [-1, 0]$ and $v_2 = [3, 4]$ form a basis for $\mathbb{R}^2$. As such, all bases for $\mathbb{R}^2$ will have 2 vectors and the dimension of $\mathbb{R}^2$ is 2.

- **Example in** $\mathbb{R}^3$. Recall, I claimed that $\{e_1, e_2, e_3\}$ forms a basis for $\mathbb{R}^3$. Verify the following.

  1. Show that e_1, e_2, and e_3 are linearly independent.
  2. Show that e_1, e_2, and e_3 span $\mathbb{R}^3$.
  3. Show that e_1, e_2, and e_3 form a basis for $\mathbb{R}^3$.
  4. Verify that the dimension of $\mathbb{R}^3$ is 3.

  Answer: Creating the matrix M whose columns consist of e_1, e_2, and e_3, $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

  Now, since $\det(M) \neq 0$, we know these vectors are linearly independent, span $\mathbb{R}^3$, and form a basis for $\mathbb{R}^3$. As such, the dimension of $\mathbb{R}^3$ is 3 (as you probably could have guessed).

Super Theorem - Version 2: This is just like version 1 with three additional properties.

Let A be an n x n matrix. The following are equivalent (TFAE)

- $Ax = b$ has a unique solution for any b.
- A is invertible (nonsingular or has an inverse).
- $A \sim I_n$. (A is row equivalent to the identity matrix).
- $\det(A) \neq 0$.
- The columns of A are linearly independent.
- The columns of A span $\mathbb{R}^n$.
- The columns of A form a basis for $\mathbb{R}^n$

It is worth emphasizing here that this theorem only applies to square matrices (n x n). A more thorough course in Linear Algebra is required to analyze the properties of non-square matrices.

Examples of Super Theorem - Version 2 in $\mathbb{R}^3$

Each of these examples gives a set of vectors in $\mathbb{R}^3$. Determine if the given vectors form a basis for $\mathbb{R}^3$. Comment on their linear independence and whether or not they span $\mathbb{R}^3$

1. $$v_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, v_3 = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$$

   Answer: Create the matrix M:

   $$M = \left[\begin{array}{c|c|c} \vdots & \vdots & \vdots \\ v_1 & v_2 & v_3 \\ \vdots & \vdots & \vdots \end{array}\right] = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 0 & -2 \\ 2 & 1 & 5 \end{bmatrix} \text{ and } \det(M) = 1(0+2) - 2(-5+4) + 1(-1-0) = 3.$$

   Here we use the Super Theorem and conclude that since $\det(M) \neq 0$, the vectors are linearly independent, they do span $\mathbb{R}^3$, and they do form a basis for $\mathbb{R}^3$.

2. $$v_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, v_3 = \begin{bmatrix} 4 \\ -2 \\ 5 \end{bmatrix}$$

Answer: Create the matrix M:

$$M = \left[\begin{array}{c|c|c} \vdots & \vdots & \vdots \\ v_1 & v_2 & v_3 \\ \vdots & \vdots & \vdots \end{array}\right] = \begin{bmatrix} 1 & 2 & 4 \\ -1 & 0 & -2 \\ 2 & 1 & 5 \end{bmatrix} \text{ and } \det(M) = 1(0+2) - 2(-5+4) + 4(-1-0) = 0.$$

Here we use the Super Theorem and conclude that since $\det(M) = 0$, the vectors are **not** linearly independent, they do **not** span $\mathbb{R}^3$, and they do **not** form a basis for $\mathbb{R}^3$. You might have noticed that $u_3 = 2u_1 + u_2$ so these vectors can not be linearly independent by the intuitive definition of linear independence.

3. $$v_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$

Answer: Since the matrix M will not result in a square matrix, we can not use the Super Theorem. However, since any basis for $\mathbb{R}^3$ must have 3 vectors we can conclude that these vectors do **not** form a basis for $\mathbb{R}^3$. Therefore, they must be linearly dependent and/or not span $\mathbb{R}^3$. We can tell that they are linearly independent because one is not a linear combination of the others (one is not a scalar multiple of the other). So, it must be that they do not span $\mathbb{R}^3$. This should make sense: You can not span a 3-dimensional space with only two vectors.

4. $$v_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, v_3 = \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix}, v_4 = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

Answer: Since the matrix M will not result in a square matrix, we can not use the Super Theorem. However, since any basis for $\mathbb{R}^3$ must have 3 vectors we can conclude that these vectors do **not** form a basis for $\mathbb{R}^3$. Therefore, they must be linearly dependent and/or not span $\mathbb{R}^3$. Determining which of these is the case is quite tricky and requires Gaussian elimination on a non-square matrix. Exploring this further requires knowing more about linear algebra such as **rank** and **nullity** of a matrix. We will not go into this but there is a good chance that these vectors will span $\mathbb{R}^3$ but are not linearly independent.

Chapter 1.5 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. For each of the sets of vectors below answer the following. 1) Determine whether the vectors are linearly independent or dependent. 2) If they are linearly dependent, find a non-zero linear combination of the vectors which results in the zero vector. 3) Finally, does the set form a basis for $\mathbb{R}^3$? You should set up these problems by hand and use MATLAB® to perform the calculations.

   (a)* $v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 2 \\ 5 \\ 11 \end{bmatrix}$

   (b) $v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 2 \\ 5 \\ 10 \end{bmatrix}$

   (c)* $v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix}$

2. Determine whether or not the given vectors form a basis for $\mathbb{R}^3$.

   (a)* $v_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}, v_3 = \begin{bmatrix} 3 \\ -4 \\ 5 \end{bmatrix}$

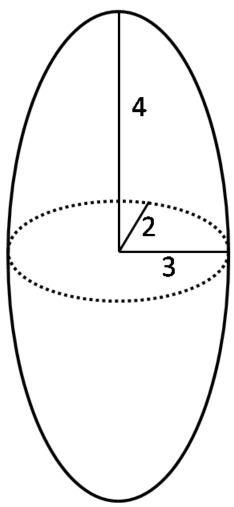
   (b) $u_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}, u_3 = \begin{bmatrix} 2 \\ 3 \\ 13 \end{bmatrix}$

1.6 Change of Basis

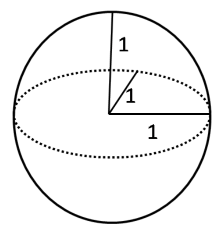
Motivation (revisited):

Suppose we have an ellipsoid with a radius vector of $[3, 2, 4]$ and want to convert this ellipsoid to a unit sphere with radius vector $[1, 1, 1]$. The idea is to find a basis for $\mathbb{R}^3$ so that the ellipsoid with respect to this basis has radius vector $[1, 1, 1]$. This is not so hard but converting everything else to this basis is the tricky part. In this sense we need to determine how to perform a **change of basis** transformation.

Ellipsoid
radius vector $= [r_x, r_y, r_z] = [3, 2, 4]$



Unit Sphere
radius vector $= [r_x, r_y, r_z] = [1, 1, 1]$



Basis Definition (revisited)

A **basis** is any minimal collection of vectors such that any vector in the vector space can be expressed as a linear combination of these vectors. We have two simple bases for $\mathbb{R}^3$ from the last section. One is the standard basis e_1, e_2, and e_3. The other is $v_1 = [3, 0, 0]^T$, $v_2 = [0, 2, 0]^T$, and $v_3[0, 0, 4]^T$ which contain the three axis radii from the ellipsoid. For example,

With respect to the standard basis the radius vector is $[3, 2, 4]$ because

$$\begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} = \mathbf{3} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \mathbf{2} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \mathbf{4} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

With respect to v_1, v_2, and v_3, the radius vector is $[1, 1, 1]$ because

$$\begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} = \mathbf{1} \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} + \mathbf{1} \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} + \mathbf{1} \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

Change of Basis

Let A and B be two different sets of basis vectors for $\mathbb{R}^n$ defined by

$$A = \{u_1, u_2, \dots u_n\} \quad \text{and} \quad B = \{v_1, v_2, \dots, v_n\}.$$

Now define the matrices U and V with these basis vectors as columns by

$$U = \left[\begin{array}{c|c|c|c} \vdots & \vdots & \vdots & \vdots \\ u_1 & u_2 & \dots & u_n \\ \vdots & \vdots & \vdots & \vdots \end{array}\right] \quad \text{and} \quad V = \left[\begin{array}{c|c|c|c} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \dots & v_n \\ \vdots & \vdots & \vdots & \vdots \end{array}\right]$$

If x is a vector in $\mathbb{R}^n$ we can write

$$x = a_1u_1 + a_2u_2 + \dots a_nu_n = Ua \quad \textbf{or} \quad x = b_1v_1 + b_2v_2 + \dots b_nv_n = Vb$$

where a is the column vector $[a_1, a_2, \dots, a_n]^T$ and $b = [b_1, b_2, \dots b_n]^T$.

Transformation Matrix

If x is a vector in $\mathbb{R}^n$, then x can be expressed in either form

$$x = Ua \quad \text{or} \quad x = Vb. \tag{1.15}$$

If we have a and want to find b we must solve the equation

$$Ua = Vb \tag{1.16}$$

for b. Since the columns of V form a basis for $\mathbb{R}^n$ it has an inverse and

$$b = V^{-1}Ua = Ta \tag{1.17}$$

The matrix $T = V^{-1}U$ is the transformation matrix from basis A to basis B and T^{-1} is the transformation matrix from basis B to basis A.

Important Observation: If $A = \{e_1, e_2, \dots, e_n\}$ is the standard basis and you want to express a vector $x = [x_1, x_2, \dots, x_n]^{\mathrm{T}}$ in terms of a different basis $B = \{v_1, v_2, \dots, v_n\}$, then equation (1.17) yields

$$b = V^{-1}Ux = V^{-1}x \tag{1.18}$$

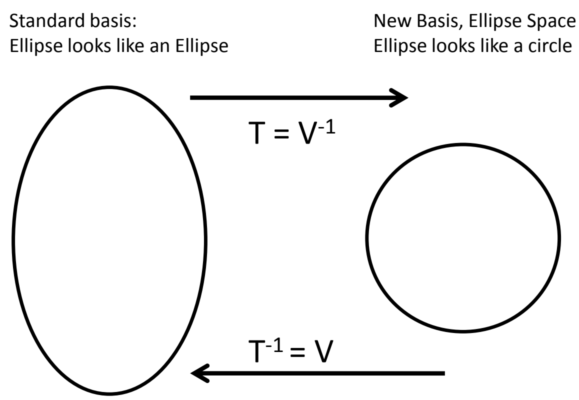
because U is just the identity matrix. Now the transition matrix T is just V^{-1}.

Therefore, converting x to a different basis is a matter of finding the inverse of the matrix with columns consisting of the vectors of the different basis. Then multiply by x.

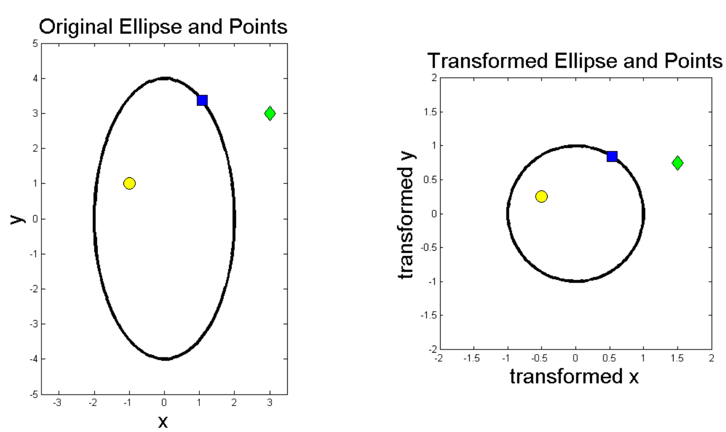
- **Example in $\mathbb{R}^2$:** Here we start with an ellipse in $\mathbb{R}^2$ with radius vector $[2, 3]$. We want to convert this into the same vector space with basis vectors $v_1 = [2, 0]$ and $v_2 = [0, 3]$. We will call the space using this basis the *ellipse space*. With this basis, our ellipse becomes a unit circle. The transformation matrix is given by equation (1.18),

$$T = V^{-1} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}.$$

So, in order to transform our ellipse into *Ellipse Space* we multiply each point of the ellipse by the transformation matrix $T = V^{-1}$. In this case, the change of basis transformation matrix is just a scaling matrix. That is, T multiplies each x and y by a scalar.



Under this transformation, points inside, on, and outside of the ellipse remain inside, on, and outside of the unit circle respectively. This gives us a way to determine whether or not the ellipse has collided with something else (code on the next page).



MATLAB® code for the Ellipse Transformation

This program transforms an ellipse into a unit circle in *Ellipse Space*. It also hits three points with the same transformation. **Important for Collision Detection:** Interior and exterior points are preserved by the transformation. This file (`Ellipse_Transform.m`) is available in the Chapter 1 program repository.

```
%% EllipseTransform.m:   Transforming an Ellipse into a Circle
clc; clf;  % Clear the command window and the figure window

%%  THE ELLIPSE AND POINTS WITH STANDARD BASIS
t = linspace(0,2*pi,100);  % t = a vector of 100 values between 0 and 2 pi.
x = 2 * cos(t);            % the x-values in the ellipse
y = 4 * sin(t);            % the y-values in the ellipse
P1 = [-1 1];    % a point inside the ellipse
P2 = [3 3];     % a point outside the ellipse
P3 = [2*cos(1) 4*sin(1)];  % a point on the ellipse

%% Plot them with respect to the standard basis
subplot(121)  % a 1x2 array of graphs - first plot
plot(x,y,'-k','linewidth',3); hold on;  %Plot the ellipse and hold it.
plot(P1(1), P1(2),'ok','MarkerFaceColor','yellow','MarkerSize',12) % plot P1
plot(P2(1), P2(2),'dk','MarkerFaceColor','green','MarkerSize',12) % plot P2
plot(P3(1), P3(2),'sk','MarkerFaceColor','blue','MarkerSize',12) % plot P3
axis equal  % keeps axis scalings equal (circles look like circles)
axis([-3.5,3.5,-5,5]);  %usage: axis([xmin, xmax, ymin, ymax]);
title('Original Ellipse and Points','fontsize',22)  % graph title
xlabel('x','fontsize',22); ylabel('y','fontsize',22)   % axis labels

%% THE Transformed ELLIPSE AND POINTS IN ELLIPSE SPACE
V = [2 0; 0 4]; % New Basis Matrix: columns from the basis in ellipse space
T= inv(V);  % The Transition matrix from standard to ellipse space.
xyTran = T*[x;y];  % Transform ellipse points into ellipse space
xTran = xyTran(1,:);  % The x-values in ellipse space
yTran = xyTran(2,:);  % The y-values in ellipse space
P1Tran = T*P1';    % Transform points into ellipse space
P2Tran = T*P2';    % Transform P2 into ellipse space
P3Tran = T*P3';    % Transform P3 into ellipse space

%% Plot them with respect to the new basis.
subplot(122) % a 1x2 array of graphs - second plot
plot(xTran,yTran,'-k','linewidth',3); hold on;  %Plot the transformed ellipse
plot(P1Tran(1), P1Tran(2),'ok','MarkerFaceColor','yellow','MarkerSize',12)
plot(P2Tran(1), P2Tran(2),'dk','MarkerFaceColor','green','MarkerSize',12)
plot(P3Tran(1), P3Tran(2),'sk','MarkerFaceColor','blue','MarkerSize',12)
axis equal  % keeps axis scalings equal (circles look like circles)
axis([-2,2,-2,2]);  %usage: axis([xmin, xmax, ymin, ymax]);
xlim([-2,2]); ylim([-2,2]);  % the x and y axis limits [(min,max)]
title('Transformed Ellipse and Points','fontsize',22)  % graph title
xlabel('transformed x','fontsize',22); ylabel('transformed y','fontsize',22)
```

Chapter 1.6 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. **Change of Basis:** Use equation (1.18) to express the given point in $\mathbb{R}^3$ with respect to a new basis consisting of the vectors: $v_1 = [1, 2, 1]^T$, $v_2 = [2, 3, 0]^T$, $v_3 = [0, 0, 2]^T$. You should set this up by hand and let MATLAB® do the calculations.

   (a)* $P = (2, 0, 3)$

   (b) $P = (1, 2, 3)$

2.* **Collision Detection 2D:** Consider a 2D ellipse centered at the origin with radius vector $[r_x, r_y] = [3, 2]$. For each point p, determine whether the point lies inside the ellipse, on the ellipse, or outside of the ellipse. Verify your answer beyond visual inspection. **Strategy:** Use a change of basis transformation (equation (1.18)) to convert the ellipse to a unit circle and express the point (x, y) in terms of the new basis. Call this new point (x_{new}, y_{new}). Let $r = x_{new}^2 + y_{new}^2$. If $r = 1$, the new point is on the unit circle and the original point is on the ellipse, if $r > 1$ the original point is outside of the ellipse, and if $r < 1$ it is inside the ellipse.

   (a)* Where does the point $p = (1.8, 1.8)$ lie with respect to the ellipse?

   (b) Where does the point $p = (-1.8, 1.3)$ lie with respect to the ellipse?

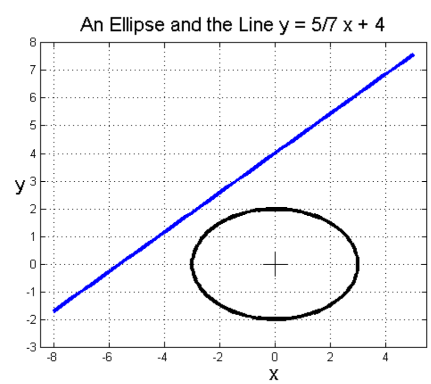
   (c) Where does the point $p = (1.50, 1.73)$ lie with respect to the ellipse?

3. **Collision Detection 3D:** Use the same strategy as above and determine if the given point lies inside, on, or outside of the ellipsoid in $\mathbb{R}^3$, centered at (0, 0, 0) with radius vector $r = [1, 2, 3]$.

   (a)* Where does the point $p = (0.5, 1.5, 2.0)$ lie with respect to the ellipsoid?

   (b) Where does the point $p = (-0.25, 1.25, 1.9)$ lie with respect to the ellipsoid?

4.* **Change of Basis Transformation:**
   Consider a 2D ellipse centered at the origin with radius vector $[r_x, r_y] = [3, 2]$, and the line $y = \frac{5}{7}x + 4$ pictured in the figure. Use a change of basis transformation to convert the ellipse to a unit circle. What does this transformation do to the line? That is, give the equation for this line in the new coordinate system. What is the slope of the new line with respect to this basis? How does it compare to the slope of the original line? What is the y-intercept with respect to the new basis? How does it compare to the original y-intercept?



   Hint: Take a couple of points from the original line, and transform these to two new points in the new basis. Now create the equation for the line from these two new points.

5. **MATLAB® - Collision Detection Assignment:**
   Get the starter program `Assignment_1_6_5_Start.m` depicted below from the Chapter 1 program repository. This program prompts the user for the x-radius and y-radius of an ellipse. It then plots this ellipse and prompts the user to click a point on the graph. You are to complete the code so that the program determines whether the point is on, inside, or outside the ellipse. Have the title of the graph state the conclusion.

```
%% Assignment_1_6_5_Start.m
% Determine if a point lies inside or outside of an ellipse.
clc; clf;clear;  % clears command window, figure, and all variables.

%% Get the user input for the ellipse using the input function.
disp('Welcome to Ellipse Collision Detection')
rx = input('what is the x-radius?');
ry = input('what is the y-radius?');

%% Draw the Ellipse
t = linspace(0,2*pi,100);         % t is the parameter to make the ellipse
x = rx * cos(t);                  % the x-values in the ellipse
y = ry * sin(t);                  % the y-values in the ellipse
plot(x,y,'-k','linewidth',2);     % Plots the ellipse.
hold on;                          % Do not erase it.
axis([-rx-1,rx+1, -ry-1,ry+1]);   % usage: axis([xmin, xmax, ymin, ymax])
axis equal      % Equal scalings on x and y axes.  Circles look like circles.

%% Get a point in the figure using ginput and plot it
disp('click a point on the graph')  % display a message to command window.
[x_point, y_point] = ginput(1);     % ginput gets a point from the graph.
plot(x_point, y_point,'ok','MarkerFaceColor','yellow')       % plot point
fprintf('your point is (%1.2f,%1.2f) \n', x_point, y_point);
disp('You take it from here and determine')
disp('if the point is inside the ellipse or not')

%% Now determine if (x_point,y_point) is inside, outside, or on the ellipse.
test_variable = 5;  % You should determine the test_variable.
if test_variable < 0.99
    title('Point is INSIDE the Ellipse')
elseif test_variable > 1.01
    title('Point is OUTSIDE the Ellipse')
else
    title('Point is ON the Ellipse')
end
```

1.7 Matrix Transformations

Here we look at how to move points and shapes around in 2 and 3 space by using matrix transformations. When I say a matrix transformation what I mean is that the new object (point, vector, geometric shape) can be obtained from the original by multiplying each point (expressed in vector form) of the object by a matrix T.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = T \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

In 2D, $[x, y]$ is the original point and $[x', y']$ is the transformed point. In 3D, $[x, y, z]$ is the original point and $[x', y', z']$ is the transformed point. The tables in this chapter list some types of transformations and their associated matrices (T). Here are some advantages of using matrix transformations.

- **A Sequence of Transformations**
  If you want to conduct a sequence of transformations represented by T_1, then T_2, and then T_3, you can combine these into a single transformation matrix

  $$T = T_3\, T_2\, T_1$$

  and then hit all of the objects in your space with this single matrix. This saves time.

- **Transforming Back**
  Suppose you transform a vector $\mathbf{v}$ into a new vector $\mathbf{v}_{new}$ by the sequence of transformations

  $$\mathbf{v}_{new} = T\mathbf{v} = T_3\, T_2\, T_1\, \mathbf{v}.$$

  Getting back to the original vector is just a matter of taking the inverse of the transformation matrix,

  $$\mathbf{v} = T^{-1}\, \mathbf{v}_{new} = T_1^{-1}\, T_2^{-1}\, T_3^{-1}\, \mathbf{v}_{new}.$$

Table 1: Scaling and Rotation Transformations

type	Description	T-Matrix (2D)	T-Matrix (3D)	Properties
Scaling	Scales along the x, y and z axes by S. We used this in transforming to *ellipse space*.	$\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$	$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}$	B
Rotation	Rotates by an angle θ **clockwise** about the origin. In 3D - rotates around the z-axis.	$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	A, B, C

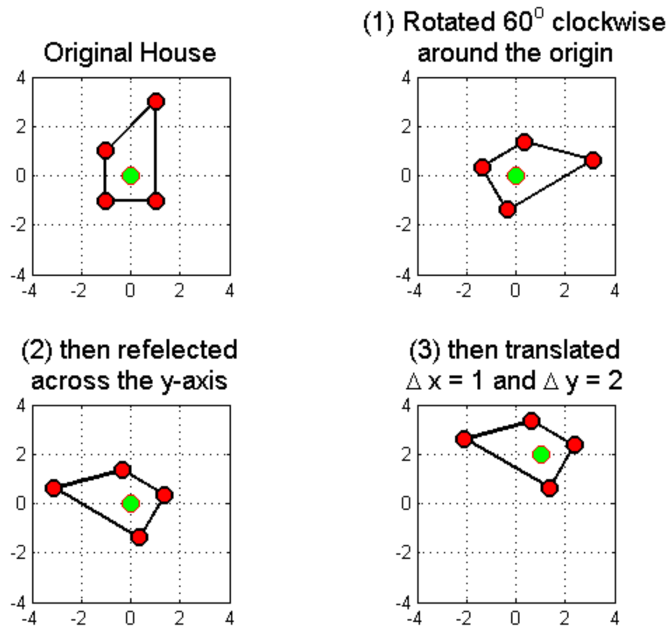
Properties Key: A: preserves angles between vectors. **B:** preserves interior and exterior points of closed curves. **C:** Preserves distances between points.

Table 2: More Matrix Transformations

type	Description	T-Matrix (2D)	T-Matrix (3D)	Properties
Reflection across y-axis.	Reflects across the y-axis. In 3D across the yz-plane.	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	A, B, C
Reflection across x-axis.	Reflects across the x-axis. In 3D across the xz-plane.	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	A, B, C
Reflection across a line.	Reflect across the line $y = mx$.	$\frac{1}{m^2+1}\begin{bmatrix} 1-m^2 & 2m \\ 2m & m^2-1 \end{bmatrix}$		A, B, C
Translations Matrix Form	Move all points a distance of $[\Delta x, \Delta y, \Delta z]$. You must *cheat* by expressing all vectors with one extra term. Put a 1 in the last position.	$\begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	A, B, C
Translations Non-Matrix Form	Move all points a distance of $[\Delta x, \Delta y, \Delta z]$. **Warning:** Not a matrix transformation.	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$	A, B, C
Horizontal Shearing	Shears (slants) all points parallel to the x-axis: $x' = x + ky$. In 3D - proportional to y.	$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & k_y & 0 \\ 0 & 1 & 0 \\ 0 & k_z & 1 \end{bmatrix}$	B
Vertical Shearing	Shears (slants) all points parallel to the y-axis: $y' = y + kx$. In 3D - proportional to x.	$\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ k_y & 1 & 0 \\ k_z & 0 & 1 \end{bmatrix}$	B

Properties Key: A: preserves angles between vectors. **B:** preserves interior and exterior points of closed curves. **C:** Preserves distances between points.

Demonstration: Here we perform three transformations in sequence by first rotating then reflecting and then translating the points of a simple *house* and the origin.



The code that generated this array of graphs (`MatrixTransforms.m`) is given on the next page.

- **Original House:** The four house-points (from bottom left clockwise) are (-1,-1), (-1,1), (1,3), (1,-1) and the origin is (0,0).
- **Rotated:** Each new point (x', y') is obtained from the original point (x, y) by

$$\begin{bmatrix} x_1' & x_2' & \dots \\ y_1' & y_2' & \dots \end{bmatrix} = \begin{bmatrix} \cos(\pi/3) & \sin(\pi/3) \\ -\sin(\pi/3) & \cos(\pi/3) \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots \\ y_1 & y_2 & \dots \end{bmatrix}$$

- **Reflected:** Each new point (x', y') is obtained from the previous point (x, y) by

$$\begin{bmatrix} x_1' & x_2' & \dots \\ y_1' & y_2' & \dots \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots \\ y_1 & y_2 & \dots \end{bmatrix}$$

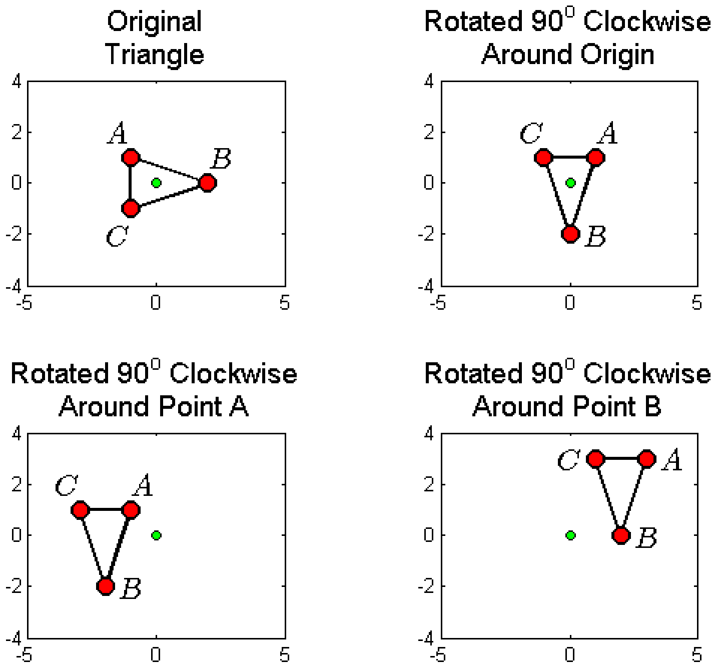
- **Translated:** We'll take advantage of MATLAB® 's ability to add a scalar to a vector (not a valid mathematical operation but handy). Each new point (x', y') is obtained from the previous (x, y) by

$$\begin{aligned} [\, x_1' \,,\, x_2' \,,\, \dots] &= [\, x_1 \,,\, x_2 \,,\, \dots] + 1 \\ [\, y_1' \,,\, y_2' \,,\, \dots] &= [\, y_1 \,,\, y_2 \,,\, \dots \,] + 2 \end{aligned}$$

MATLAB® code: Here is the code for rotating, reflecting, and translating the *house* from the previous page. The file, `Matrix_Transforms.m`, is available in the Chapter 1 program repository.

```
%% This is Matrix_Transforms.m
% Tranforming shapes in 2-space
clc; clf;    % clear command window; clear figure;
xHouse = [-1 -1 1  1 -1];  % x-values of the house points
yHouse = [-1  1 3 -1 -1];  % y-values of the house points
P = [0 0]';                % A point at the origin

%% original graph
subplot(221)     % in a 2x2 array of graphs, start the 1st graph
plot(xHouse,yHouse,'k-o','linewidth',2,'MarkerSize',10,'MarkerFace','r')
hold on; grid on; axis equal; axis([-4,4,-4,4]);
plot(P(1),P(2),'ro','MarkerSize',10,'MarkerFace','g')
title('Original House','fontsize',14)

%% Rotate it 60 degrees
subplot(222) % in a 2x2 array of graphs, start the 2nd graph.
ThetaDegrees = 60;
theta = ThetaDegrees*pi/180;  % CONVERT DEGREES TO RADIANS!!
T1 = [cos(theta) sin(theta); -sin(theta) cos(theta)];
xyHouse1 = T1*[xHouse;yHouse];
xHouse1 = xyHouse1(1,:); yHouse1 = xyHouse1(2,:);
P1 = T1*P;
plot(xHouse1,yHouse1,'k-o','linewidth',2,'markersize',10,'markerfac','r')
hold on; grid on; axis equal; axis([-4,4,-4,4]);
plot(P1(1),P1(2),'ro','markersize',10,'markerface','g')
%title('Rotated 60^o clockwise','fontsize',14)
title({'(1) Rotated 60^o clockwise';'around the origin'},'fontsize',14)
%% Reflect Across the y-axis
subplot(223)  % in a 2x2 array of graphs, start the 3rd graph
T2 = [-1 0; 0 1];
xyHouse2 = T2*[xHouse1;yHouse1];
xHouse2 = xyHouse2(1,:); yHouse2 = xyHouse2(2,:);
P2 = T1*P1;
plot(xHouse2,yHouse2,'k-o','linewidth',2,'markersize',10,'markerface','r')
hold on; grid on; axis equal; axis([-4,4,-4,4]);
plot(P2(1),P2(2),'ro','markersize',10,'markerfac','g')
title({'(2) then refelected';'across the y-axis'},'fontsize',14)

%% Translated by [dx,dy] = [1,2]
subplot(224) % in a 2x2 array of graphs, start the 4th graph
dx = 1; dy = 2;
xHouse3 = xHouse2 + dx;  yHouse3 = yHouse2 + dy;
P3 = P2 + [dx dy]';
plot(xHouse3,yHouse3,'k-o','linewidth',2,'markersize',10,'markerface','r')
hold on; grid on; axis equal; axis([-4,4,-4,4]);
plot(P3(1),P3(2),'ro','markersize',10,'markerfac','g')
title({'(3) then translated';'\Delta x = 1 and \Delta y = 2'},'fontsize',14)
```

Rotations: Rotations around the origin are easy. Rotating around another point is a little trickier. This requires a translation that brings the rotation point to the origin, then rotate around the origin, and then translate back. The order of this sequence is very important. Below, a triangle is rotated by 90° around various points. The process for each rotation is described below the graphs.



The code that generated this array of graphs (`Rotations.m`) is in the Chapter 1 program repository.

- **Original Triangle:** The three triangle-points are A(-1,1), B(2,0), C(-1,-1).
  Each new point (x', y') is obtained from the original point (x, y) by the equations below.
- **Rotated 90° Clockwise Around the Origin:**

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\pi/2) & \sin(\pi/2) \\ -\sin(\pi/2) & \cos(\pi/2) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

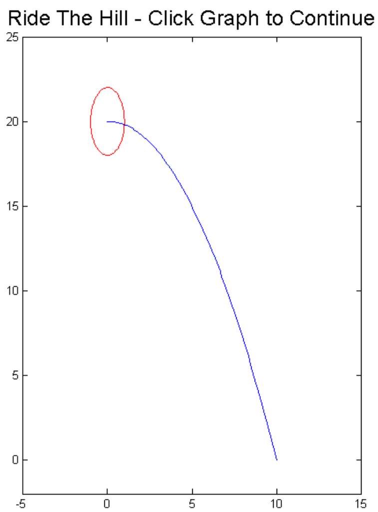
- **Rotated 90° Clockwise Around A(-1,1):**

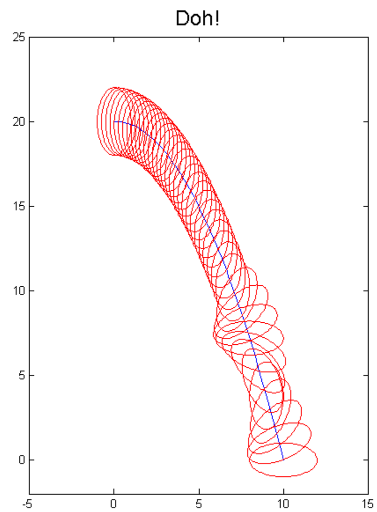
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\pi/2) & \sin(\pi/2) \\ -\sin(\pi/2) & \cos(\pi/2) \end{bmatrix} \begin{bmatrix} x-(-1) \\ y-(1) \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

- **Rotated 90° Clockwise Around B(2,0):**

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\pi/2) & \sin(\pi/2) \\ -\sin(\pi/2) & \cos(\pi/2) \end{bmatrix} \begin{bmatrix} x-(2) \\ y-(0) \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

<u>**Animations:**</u> Here we graph a *'guy'* (just an ellipse) skiing down a hill. Halfway down, he falls and starts to roll down the hill. Here are some clips from this simple animation:





Game Plan

1. **The Hill:** I just took the downward parabola $y = -x^2$ and made some alterations.

   starting graph: $y = -x^2$

   bump it up 100 units: $y = 100 - x^2$

   too tall and skinny - squish it down: $y = .2\left(100 - x^2\right) \quad \text{for} \quad x \in [0, 10]$

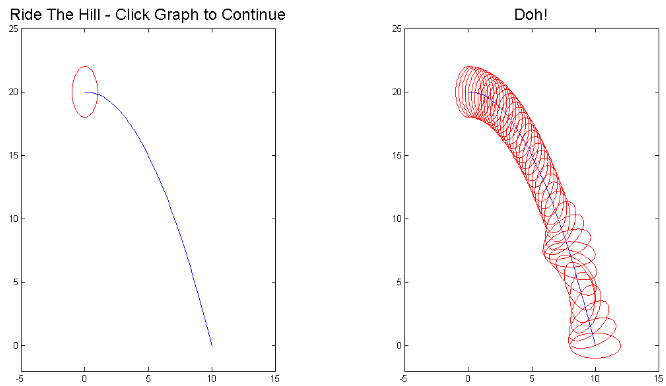
   MATLAB® Note: When you want x^2 but x is a vector, you probably want to square every term in x. In MATLAB® this is accomplished by using a period before the operation. So, to square every term in the vector x you use the command:

   ```
   y = x.^2
   ```

2. **The Guy:** Now we'll define the 'guy' as an ellipse twice as tall as he is wide. To define the set of x and y values for this ellipse, we start with a parameter t which we will let range from 0 to 2π. So the x and y values are defined by

   $$x = \cos(t) \quad y = 2\cos(t) \quad \text{for} \quad t \in [0, 2\pi]$$

   Unfortunately this ellipse is centered at (0,0). We'll worry about moving (translating) him next.



3. Now we have to move our 'guy' along the trajectory. We'll do this with a simple translation. For each frame of the movie, we'll translate each x-value of the ellipse by $dx = x$, and each y-value by $dy = y(x)$.

4. When the *guy* gets about half-way down the hill (at $y = 10$), we want to start rotating him in the clockwise direction. **IMPORTANT:** We must rotate him before the translation otherwise the translated guy will rotate around the origin and that will be bad.

5. Now we have to code it. The complete code is given on the next page. But there are some specifics worth noting.

   - If you want to write a quick function in MATLAB® , a good choice is something called an *anonymous* function such as the one in our program:

     ```
     F = @(x)[.2*(100-x.^2)]
     ```

     Here, `F` is the name of the function, the @ symbol is the command that MATLAB® recognizes, the first set of parentheses contains the independent (input) variable(s), and the square brackets contain the dependent (return) variable(s).

   - `linspace(a,b,n)` creates a vector of `n` values starting at `a` and ending at `b`. The values are equally spaced. You can create vectors like this by hand, but this function makes it so much easier.

   - In the animation it is best to give each graph a name so that you can delete the desired objects as needed. This way you do not have to erase and redraw the entire environment for each frame.

   - Don't be afraid to change the `pause` time. Depending the number of frames you might want to increase or decrease this value. If you have no pauses, the animation will go so fast that you might not even see it. Then you could spend forever trying to figure out what's wrong.

6. Read through the code on the next page. Make sure you understand what each line does and why we need it. Minor deviations can cause major problems.

MATLAB® Code for the Wipe-Out Animation available in the Chapter 1 program repository.

```
%%   This is Wipe_out.m
clc; clf; clear; % clear command window, figure, and all variables

%% Make the hill.
F = @(x)[.2*(100 - x.^2)];      % Defines an 'anonymous' function of x called F.
x_hill = linspace(0,10,200);    % 200 x-values of the hill graph from 0 to 10.
y_hill = F(x_hill);             % y-values of the hill graph

%% Make the guy.
t = linspace(0,2*pi,100);          % 100 t-values for the ellipse from 0 to 2 pi
x_guy = cos(t); y_guy = 2*sin(t);  % The x and y values for the ellipse (guy).

%% Original plot with guy on top of hill
hill_plot = plot(x_hill,y_hill,'-b');   % plot the hill.
axis equal; hold on;                    % make axes equal and don't erase.
guy_plot = plot(x_guy, y_guy+20, '-r'); % plot the guy on top of the hill.
axisbounds = [-5,15,-2,25];             % axis limits [xmin,xmax,ymin,ymax].
axis(axisbounds);                       % set the axis bounds.
title('Ride The Hill - Click Graph to Continue','fontsize',18) % title

%% Prompt to start animation
w = waitforbuttonpress;  % waits for a button push or mouse click.
title('Ride On!','fontsize',18)        % change the title.

%% Initial Rotation Angle
theta = 0;               % original rotation angle.

%% Start the Animation.
for dx = 0:.1:10         % x goes from 0 to 10 by steps of .1
    dy = F(dx);          % sets dy to follow the hill
    delete(guy_plot);    % deletes the plot of the guy called guy_plot.
    if dy > 10;          % Dont' fall until dy < 10.
        guy_plot = plot(x_guy+dx, y_guy+dy, '-r'); % new guy_plot
    else                 % If the guy goes below y = 10, he starts to rotate.
        title('Doh!','fontsize',18)
        theta = theta + pi/8;  % increment the rotation angle theta.
        T_rotate = [cos(theta) sin(theta); -sin(theta) cos(theta)];
                   % T_rotate is the rotation matrix
        rotated_guy = T_rotate*[x_guy;y_guy];  % Rotate all points in guy.
        rotated_x_guy = rotated_guy(1,:);      % get the x-values.
        rotated_y_guy = rotated_guy(2,:);      % get the y-values.
        guy_plot = plot(rotated_x_guy + dx, rotated_y_guy + dy, '-r');
                   % above = new guy_plot rotated then translated
    end   % end of 'if' statement
    axis(axisbounds);  % set the axis bounds.
    pause(.01);        % slow down the animation by pausing .01 seconds
end      % End of animation loop
```

Chapter 1.7 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. Suppose the point (0,3) is translated 2 units up (in the y-direction) and 1 unit to the right (in the x-direction) and then rotated counter-clockwise around the origin by 60^o. Do the set up by hand and perform the calculations in MATLAB® .

   (a)* Where would this point go? Round your answer to one decimal place.

   (b) Suppose you did these transformations in the reverse order. Now where would this point go?

2. Suppose the point (-1,3) is first rotated clockwise around the origin by $\theta = \dfrac{\pi}{6}$ radians and then reflected across the x-axis. Do the set up by hand and perform the calculations in MATLAB® .

   (a)* Where would this point go? Round your answer to one decimal place.

   (b) Suppose you did these transformations in the reverse order. Now where would this point go?

3. Suppose the point (x, y) is first rotated counter-clockwise around the origin by 45^o and then translated 2 units left and 3 units up. After this sequence of transformations, the point (x,y) is now (0,5).

   (a)* What was the original point (x, y)? Round your answer to one decimal place.

   (b) Suppose the transformations were done in reverse order. What would the original point be?

4. Suppose the point (x, y) is first rotated clockwise around the origin by $\theta = \dfrac{\pi}{6}$ radians and then reflected across the x-axis. After this sequence of transformations, the point (x,y) is now (-2,3).

   (a)* What was the original point (x, y)? Round your answer to two decimal places.

   (b) Suppose the transformations were done in reverse order. What would the original point be?

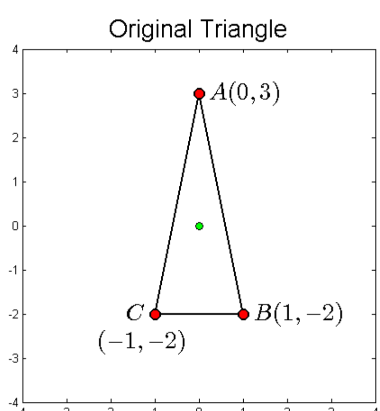
5. **Rotations:**
   Consider the triangle depicted to the right. Determine what happens to point C after

   (a)* rotating the triangle 45° counter-clockwise around the origin.

   (b)* rotating the triangle 45° counter-clockwise around point A.

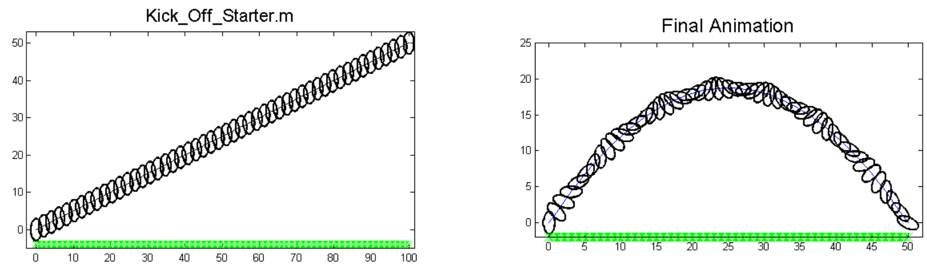
   (c) rotating the triangle 45° counter-clockwise around point B.

   (d) rotating the triangle 45° counter-clockwise around point C.



6. **(MATLAB®) Kick-Off!**

   In this assignment you are to create an animation of a football being kicked across a trajectory that starts at (0,0) and ends somewhere down the field at the point $(x_{end}, 0)$. The football should be an ellipse that rotates counter-clockwise as it moves along a smooth trajectory. Below (right) is what it should look like in the final animation. You can figure out the trajectory below or create your own. There is a starter program called `Kick_Off_Starter.m` in the program repositorythat will help get you started. It produces the animation below (left).



   Required Features:

   (a) The ball must look like a football (ellipse) and remain the same shape throughout the animation using `axis equal;`

   (b) The axes must remain constant throughout the animation by using `axis([`x_{min}`, `x_{max}`, `y_{min}`, `y_{max}`])` after the `axis equal;` command.

   (c) The ball must start at the origin and land on the ground.

   (d) The trajectory must be visible in the animation.

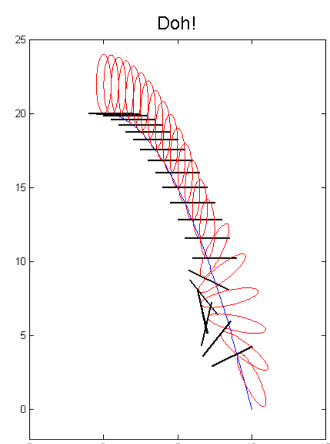
   (e) The trajectory must be smooth. (no sharp turns).

   (f) The ball must rotate counter-clockwise as it follows it's trajectory.

7. **(MATLAB®) Improved Wipe Out:**

   Take the skiing program (`Wipe_out.m`) and alter it for a better animation. Two good places to start are:

   - Make the guy (ellipse) stand on the ground. Currently the middle of the ellipse is at ground level. That looks bad.
   - Give him skis that rotate appropriately after he wipes out.
   - These changes should produce an animation similar to the one depicted.



Chapter 1 Project: Real-Time Collision Detection

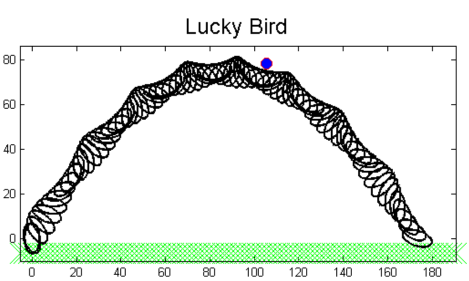
Here we combine the collision detection from Chapter 1.6 and the matrix transforms/animations from Chapter 1.7 to perform what is known as *Real-Time Collision Detection*.

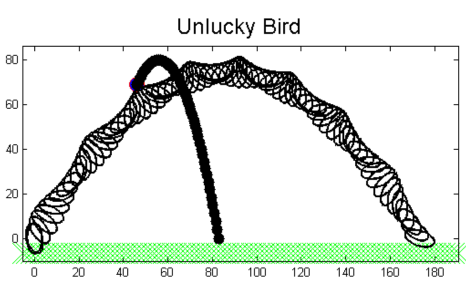
Summary: You are to create a football kick-off animation. Before the ball is set in motion, you must prompt the user to place a bird somewhere on the plane. Then the ball is to fly along it's trajectory and the program must determine if the bird gets hit by the ball. You do this by performing a collision detection at every rendering of the figure. This is called *Real-Time Collision Detection*. If there is a collision, the bird should fall to the ground in some fashion while the ball continues on its trajectory. If there is no collision the bird should remain in it's initial location throughout the animation. The starter-code (`Project_1_Starter.m`) is available in the Chapter 1 program repository.

Required Features:

1. The ball must start by resting on the ground at the (0,0) location in the vertical position.
2. The trajectory must be visible by the viewer at this screen-shot.
3. The user is then prompted to place a bird on the figure by clicking on a point. The bird will remain stationary. The bird should just be a visible point on the graph.
4. When the ball is kicked. It must rotate counter-clockwise as it follows it's trajectory.
5. The ball must maintain its shape and the axes must remain constant throughout the animation with `axis equal;` followed by `axis([`x_{min}`, `x_{max}`, `y_{min}`, `y_{max}`])`
6. Erase the trajectory curve once the ball starts on its way.
7. At every screen-shot (or properly chosen subset) you must determine if the ball has hit the bird.
8. If the bird is hit, it should fall to the ground in some fashion while the ball continues on its trajectory. The title of the graph should mention this.
9. If the bird is never hit, it should remain in its initial location throughout the flight of the ball.

Here are some screen-shots from a miss and a hit.

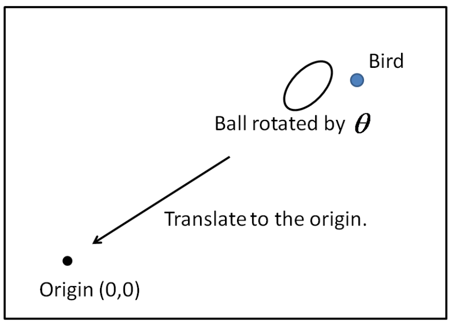




Hints

- **Step size matters:** Make sure the animation loop takes small enough steps so that the ball does not go through the bird without a collision being detected. Don't make the steps so small that each pass during the debugging process takes forever.

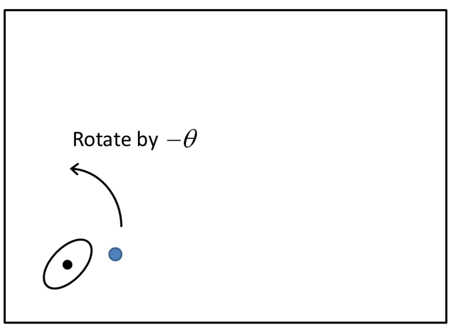
- **Perform the collision detection back at the origin**. We have only done collision detection with a non-rotated ellipse centered at the origin. So we have to perform the collision detection in that scenario. Before every frame is made, the ball starts like that anyway. So, we have to translate the bird back to the origin and un-rotate it. **The order makes all the difference.**



Assume the ball is currently centered at (x_{ball}, y_{ball}) and rotated by θ.

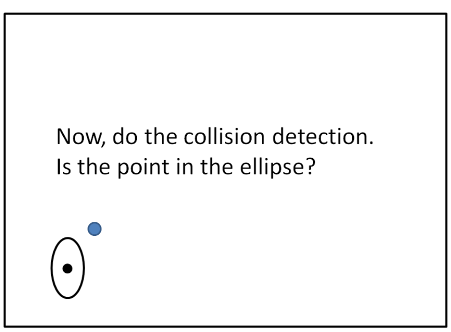
Perform the collision detection back at the origin, keeping the relative positions the same.

The bird's translated position will be
$(x_{bird} - x_{ball}, y_{bird} - y_{ball})$.
The ball starts centered at the origin anyway.



The ball will be rotated by θ.
You need to **un-rotate** the situation here.

So, rotate the bird by $-\theta$.
The ball starts in the un-rotated position anyway.



Perform the collision detection
as in Chapter 1.6

Chapter 2

Vectors and the Geometry of Space

In this chapter we formalize our study of the mathematical geometry of space. It starts with some material that should be familiar to you from the last chapter. Specifically, we start with vectors. Previously when we discussed vectors, we considered a vector as an ordered list of numbers. We demonstrated how these vectors can be used to describe points in 2 and 3 dimensional space. Now we will consider a vector as a directed line segment that has a length and a direction. This vector can be situated anywhere in 2 or 3 space. As such, a single vector actually describes infinitely many possible directed line segments starting at any location in our geometry. Because of this ambiguity we generally consider the vector to start at the origin. Once this assumption is made, our definition of a vector results in our previous notion of a specific point in 2 or 3 space. By creating a vector-valued function we can trace out the endpoints of these vectors which makes a curve in space. We introduced dot products in the last chapter. In this chapter we will expand on this and describe what a dot product produces from a geometric perspective. We also introduce a different type of vector product called the cross product and describe this in geometric terms. Once we have these operations in our toolbox, we can go on to describe lines and planes in 3-space and investigate how and when these geometric objects intersect.

2.1 Vectors in the Plane

In this section we investigate vectors in the context of 2-dimensional geometry (vectors in the plane). We start with a couple definitions.

- A **directed line segment** from an initial point P to a terminal point Q is denoted $\overrightarrow{PQ}$. It has a length (or magnitude) denoted by $||\overrightarrow{PQ}||$. Directed line segments with the same length and direction are called **equivalent**. For any directed line segment, there are infinitely many equivalent directed line segments.

- A **vector** is a standardized representation of all equivalent directed line segments.

Component Form of a Vector

If $\overrightarrow{v}$ is a vector with initial point at the origin (0,0) and terminal point $(x,\ y)$ then the component form of $\overrightarrow{v}$ is $\overrightarrow{v} = \langle x, y\rangle$. If $\overrightarrow{v}$ is a vector defined by the directed line segment with initial point $P = (P_1,\ P_2)$ and terminal point $Q = (Q_1,\ Q_2)$, then the component form of this vector is defined by $\overrightarrow{v} = \langle Q_1 - P_1, Q_2 - P_2\rangle$. This has the geometric effect of taking the original directed line segment and translating it to an equivalent one with initial point at the origin.

- **Example:** Given $P = (-1,\ 3)$ and $Q = (4,\ -5)$, find $\overrightarrow{v} = \overrightarrow{PQ}$.

  Answer: The vector $\overrightarrow{PQ}$ =Q - P= $\langle 4 - (-1),\ -5 - 3\rangle = \langle 5,\ -8\rangle = \overrightarrow{v}$.

Notation

Vectors are denoted in two different ways. In typeset material a vector is generally denoted by a lower case, bold-faced letter such as **u**, **v**, or **w**. When written by hand, the arrow notation is used such as $\overrightarrow{u}$, $\overrightarrow{v}$, or $\overrightarrow{w}$. Bold-faced notation is primarily used in this text but the arrow notation is used in some situations involving a vector representation of a directed line segment such as $\overrightarrow{PQ}$. Components of a vector are generally given in terms of the vector variable, such as

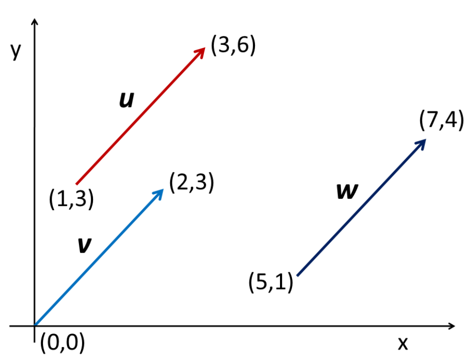
$$\mathbf{u} = \langle u_1,\ u_2\rangle \qquad \text{and} \qquad \mathbf{v} = \langle v_1,\ v_2\rangle\,.$$

Equivalent Vectors:

Two vectors are equivalent if they have the same length and direction. This results in the two vectors having identical terms when written in component form. Thus, if $\mathbf{u} = \langle u_1,\ u_2\rangle$ and $\mathbf{v} = \langle v_1,\ v_2\rangle$, then

$$\mathbf{u} = \mathbf{v} \quad \Longleftrightarrow \quad u_1 = v_1 \text{ and } u_2 = v_2. \tag{2.1}$$

- **Example:** Verify that the three vectors in the figure below are equivalent.



Answer:
The vectors in component form are

$\mathbf{u} = \langle 3 - 1,\ 6 - 3\rangle = \langle 2,\ 3\rangle$,
$\mathbf{v} = \langle 2 - 0,\ 3 - 0\rangle = \langle 2,\ 3\rangle$,
$\mathbf{w} = \langle 7 - 5,\ 4 - 1\rangle = \langle 2,\ 3\rangle$.

Since the components are identical these vectors are equivalent.

Scalar Multiplication of Vectors

If k is a scalar (real number) and $\mathbf{u} = \langle u_1,\ u_2 \rangle$ is a vector, then

$$k\,\mathbf{u} = \langle ku_1,\ ku_2 \rangle\,. \tag{2.2}$$

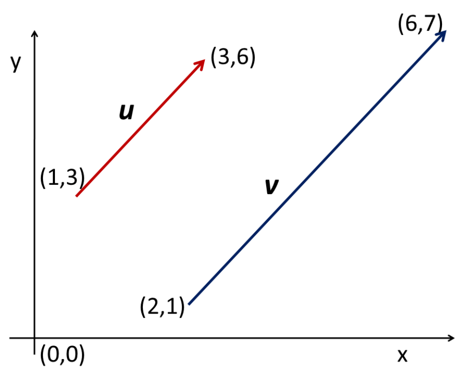
From a geometric perspective, this multiplies the length of $\mathbf{u}$ by $|k|$. If k is negative, then the direction of $\mathbf{u}$ switches. If $k = -1$, $k\,\mathbf{u}$ looks just like $\mathbf{u}$ only pointing in the opposite direction.

Parallel Vectors

Two vectors are parallel if they are scalar multiples of each other. If $\mathbf{u} = \langle u_1,\ u_2 \rangle$ and $\mathbf{v} = \langle v_1,\ v_2 \rangle$, then

$$\begin{aligned} \mathbf{u} \text{ is parallel to } \mathbf{v} \quad &\iff \quad \mathbf{v} = k\,\mathbf{u} \quad \text{for some scalar } k, \\ &\iff \quad v_1 = k\,u_1 \text{ and } v_2 = k\,u_2. \end{aligned} \tag{2.3}$$

- **Example:** Verify that the two vectors in the figure below are parallel.



Answer:
The vectors in component form are
$\mathbf{u} = \langle 3-1,\ 6-3 \rangle = \langle 2,\ 3 \rangle$
$\mathbf{v} = \langle 6-2,\ 7-1 \rangle = \langle 4,\ 6 \rangle$.

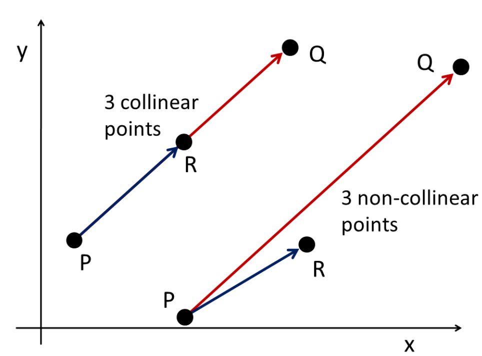
Notice that

$$\frac{v_1}{u_1} = \frac{4}{2} = 2 \text{ and } \frac{v_2}{u_2} = \frac{6}{3} = 2,$$

so $\mathbf{v} = 2\mathbf{u}$, and the vectors are parallel.

Collinear Points

To determine if three points P, Q, and R are collinear (lie in a line), check to see whether the vectors $\overrightarrow{PQ}$ and $\overrightarrow{PR}$ are parallel. If they are, then the three points are collinear as in the figure to the right.

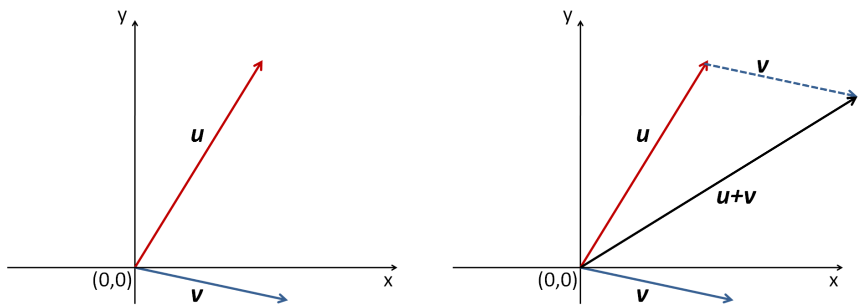


Example: Verify that the points
$A(1,\ 3)$, $B(5,\ 6)$, and $C(9,\ 9)$ are collinear.

Answer:
$\overrightarrow{AB} = \langle 5-1,\ 6-3 \rangle = \langle 4,\ 3 \rangle$
$\overrightarrow{AC} = \langle 9-1,\ 9-3 \rangle = \langle 8,\ 6 \rangle$.
Since $\overrightarrow{AC} = 2\,\overrightarrow{AB}$, these vectors are parallel and the points are collinear.

Vector Addition If $\mathbf{u} = \langle u_1,\ u_2 \rangle$ and $\mathbf{v} = \langle v_1,\ v_2 \rangle$, then

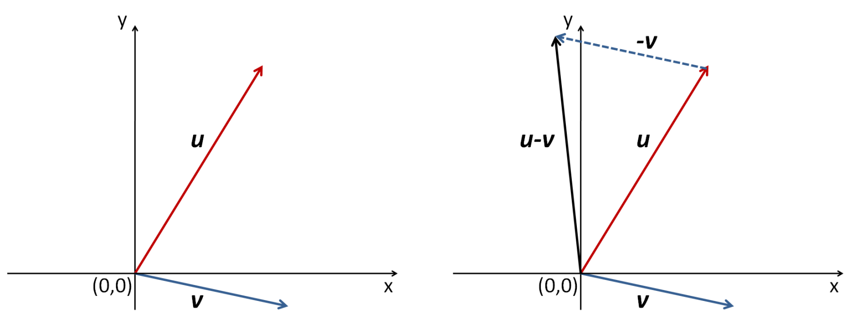
$$\mathbf{u} + \mathbf{v} = \langle u_1 + v_1,\ u_2 + v_2 \rangle \,. \tag{2.4}$$



Geometric Interpretation (Tip-to-Tail): Align the tip of $\mathbf{u}$ with the *tail* of $\mathbf{v}$ and then connect the tail of $\mathbf{u}$ with the tip of $\mathbf{v}$. If $\mathbf{u}$ and $\mathbf{v}$ represent forces, then $\mathbf{u} + \mathbf{v}$ is called the **resultant force**.

Vector Subtraction If $\mathbf{u} = \langle u_1,\ u_2 \rangle$ and $\mathbf{v} = \langle v_1,\ v_2 \rangle$, then

$$\mathbf{u} - \mathbf{v} = \langle u_1 - v_1,\ u_2 - v_2 \rangle \,. \tag{2.5}$$



Geometric Interpretation (Tip-to-Tip): Align the tip of $\mathbf{u}$ with the tip of $\mathbf{v}$, then connect the tail of $\mathbf{u}$ to the tail of $\mathbf{v}$. It might be easier just to picture $\mathbf{u} - \mathbf{v}$ as $\mathbf{u} + (-\mathbf{v})$.

Length (Magnitude or Norm) of a Vector

A vector in component form $\mathbf{v} = \langle v_1,\ v_2 \rangle$ has length (or magnitude, or norm) given by

$$||\mathbf{v}|| = \sqrt{v_1^2 + v_2^2}\,. \tag{2.6}$$

A vector of zero length is called the **zero vector**, $\mathbf{0} = \langle 0,\ 0 \rangle$.

- **Example:** Find the length of the vector from $P = (-1,\ 3)$ to $Q = (4,\ 15)$.
  Answer: $\overrightarrow{PQ} = \langle 5, 12 \rangle$, and $||\overrightarrow{PQ}|| = \sqrt{5^2 + 12^2} = \sqrt{169} = 13$.

Unit Vectors and Normalizing a Vector

A **unit vector** is a vector with length 1. The unit vector $\mathbf{u}$ in the direction of $\mathbf{v}$ is given by

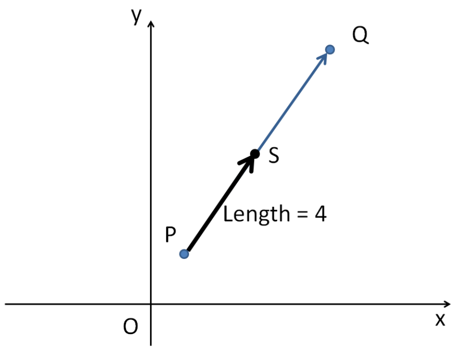
$$\mathbf{u} = \frac{1}{||\mathbf{v}||}\,\mathbf{v} = \frac{\mathbf{v}}{||\mathbf{v}||}. \tag{2.7}$$

This is called **normalizing** the vector $\mathbf{v}$.

- **Example:** Find the unit vector ($\mathbf{u}$) in the direction of $\mathbf{v} = \langle -3,\ 2\rangle$. That is, normalize $\mathbf{v}$.

  Answer: $||\mathbf{v}|| = \sqrt{(-3)^2 + 2^2} = \sqrt{13}$. So, $\mathbf{u} = \frac{1}{\sqrt{13}}\mathbf{v} = \left\langle \frac{-3}{\sqrt{13}}, \frac{2}{\sqrt{13}} \right\rangle$

- **Application:** Given the points $P(2,\ 3)$ and $Q(7,\ 12)$, find point S such that S is 4 units from P in the direction of Q.



Strategy: Normalize $\overrightarrow{PQ}$ and multiply it by 4 to get $\overrightarrow{PS}$, then *add*** $\overrightarrow{PS}$ to the point P.

$\overrightarrow{PQ} = \langle 7-2,\ 12-3\rangle = \langle 5,\ 9\rangle$

$||\overrightarrow{PQ}|| = \sqrt{5^2+9^2} = \sqrt{106}$

$\mathbf{u} = \frac{1}{||\overrightarrow{PQ}||}\overrightarrow{PQ} = \frac{1}{\sqrt{106}}\langle 5,\ 9\rangle$

$\overrightarrow{PS} = \frac{4}{\sqrt{106}}\langle 5, 9\rangle$

$S = P + \overrightarrow{PS} = (2,3) + \frac{4}{\sqrt{106}}\langle 5,9\rangle \approx (3.94, 6.50)^{**}$

** Mathematically speaking, you can't add a vector to a point (without defining a new operation), but we allow it here for practical purposes.

Vector Properties Let $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ be vectors and let c, d, and k be scalars.

(1) Commutative: $\mathbf{u}+\mathbf{v} = \mathbf{v}+\mathbf{u}$.

(2) Associative: $\mathbf{u}+(\mathbf{v}+\mathbf{w}) = (\mathbf{u}+\mathbf{v})+\mathbf{w}$.

(3) Additive Identity: $\mathbf{0} = \langle 0,0\rangle$ is called the zero vector and $\mathbf{u}+\mathbf{0} = \mathbf{u}$.

(4) Additive Inverse: $\mathbf{u}+(-\mathbf{u}) = \mathbf{0}$.

(5) $c\,(d\,\mathbf{u}) = c\,d\,\mathbf{u}$.

(6) Distributive Properties:

- $(c+d)\,\mathbf{u} = c\,\mathbf{u} + d\,\mathbf{u}$
- $c(\mathbf{u}+\mathbf{v}) = c\,\mathbf{u} + c\,\mathbf{v}$

(7) $1\,\mathbf{u} = \mathbf{u}$ and $0\,\mathbf{u} = \mathbf{0}$

(8) $||k\,\mathbf{u}|| = |k|\,||\mathbf{u}||$

(9) Triangle inequality: $||\mathbf{u}+\mathbf{v}|| \le ||\mathbf{u}|| + ||\mathbf{v}||$

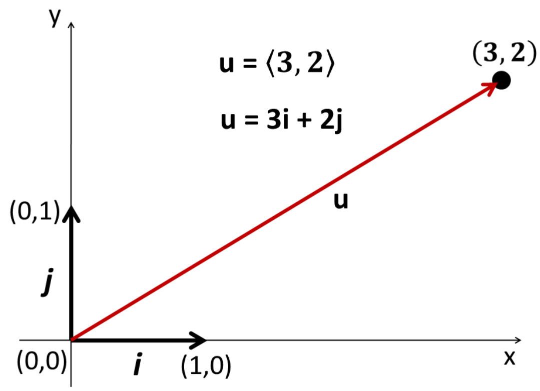
The Standard Unit (Basis) Vectors in 2D

The standard unit vectors (basis vectors) in 2-space are $\mathbf{i} = \langle 1,\ 0\rangle$ and $\mathbf{j} = \langle 0,\ 1\rangle$.

Any vector $\mathbf{u} = \langle u_1,\ u_2\rangle$ can be expressed as a linear combination of $\mathbf{i}$ and $\mathbf{j}$ by

$$\mathbf{u} = u_1\,\mathbf{i} + u_2\,\mathbf{j}.$$

Here, u_1 is called the **horizontal** component of $\mathbf{u}$ and u_2 is called the **vertical** component of $\mathbf{u}$.



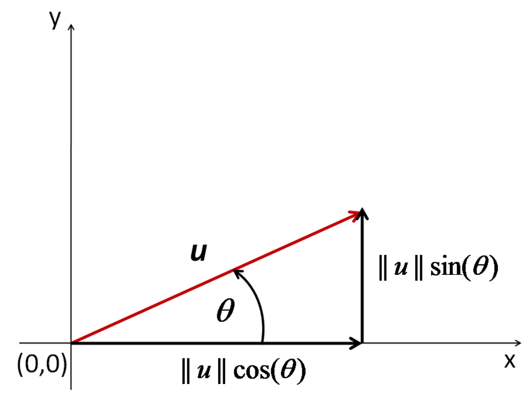
- **Example:** Express the vector $\mathbf{u} = \langle 2,\ 3\rangle$ as a linear combination of the standard unit vectors.

  Answer: $\mathbf{u} = \langle 2,\ 3\rangle = 2\,\mathbf{i} + 3\,\mathbf{j}$. Yes, it's that simple.

Polar Representation of Vectors

If $\mathbf{u}$ is a vector with length $||\mathbf{u}||$ that makes a (counter-clockwise) angle θ from the positive x-axis, then

$$\mathbf{u} = ||\mathbf{u}||\cos\theta\ \mathbf{i} + ||\mathbf{u}||\sin\theta\ \mathbf{j} = ||\mathbf{u}||\ \langle\cos\theta,\ \sin\theta\rangle\,.$$



- **Example:** Suppose the vector $\mathbf{u}$ has length 2 and makes an angle of 60^o with the positive x-axis. Express $\mathbf{u}$ as a linear combination of $\mathbf{i}$ and $\mathbf{j}$, and then give the component form of $\mathbf{u}$.

  Answer: Since the vector must have length 2, we know that $||\mathbf{u}|| = 2$. Also, $60^o = \pi/3$ radians.

  Expressed as a linear combination of $\mathbf{i}$ and $\mathbf{j}$,

  $\mathbf{u} = ||\mathbf{u}||\ \cos\theta\ \mathbf{i} + ||\mathbf{u}||\ \sin\theta\ \mathbf{j}\ =\ 2\ \cos(\pi/3)\ \mathbf{i} + 2\ \sin(\pi/3)\mathbf{j}\ =\ 2\,\frac{1}{2}\,\mathbf{i} + 2\,\frac{\sqrt{3}}{2}\,\mathbf{j}\ =\ \mathbf{i} + \sqrt{3}\,\mathbf{j}.$

  Expressed in component form, $\mathbf{u} = \left\langle 1,\ \sqrt{3}\right\rangle$.

Chapter 2.1 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. The initial point and terminal points of a directed line segment are given. (1) Sketch the directed line segment. (2) Write the vector in component form. (3) Sketch the vector with its initial point at the origin.

   (a)* Initial Point: (1,3) Terminal Point: (-2, -7).

   (b) Initial Point: (1,3) Terminal Point: (2, 7).

2. Find the vectors $\mathbf{u} = \overrightarrow{PQ}$ and $\mathbf{v} = \overrightarrow{RS}$ for the given points P, Q, R and S. Then (1) determine if **u** and **v** are equivalent vectors and (2) determine if **u** and **v** are parallel vectors.

   (a)* **u**: $P(-2,3)$, $Q(-4,-2)$ **v**: $R(2,4)$, $S(-2,3)$.
   (b) **u**: $P(0,2)$, $Q(3,0)$ **v**: $R(-1,5), S(2,3)$.
   (c)* **u**: $P(2,2)$, $Q(5,0)$ **v**: $R(1,3), S(-5,7)$.

3. Use vectors to determine whether the points are collinear. Plot the points by hand or in MATLAB® to check your answers.

   (a)* P(0, -2), Q(3, 4), R(2, 2)
   (b) P(1, 2), Q(2, 5), R(4, 8)

4. (**MATLAB®**) Write a program that plots two points on the plane. Now prompt the user to click somewhere on the graph. The program must determine whether the third point lies on the line created by the initial two points. This is far more complicated than you might think because remember: **you can't divide by zero**. Now, it will be almost impossible to click a point exactly on the line. How could you include a *close enough* option?

5.* Find the value of w needed to make the points P, Q, and R collinear.

   (a)* $P(-1,3)$ $Q(2,7)$ $R(7,w)$
   (b) $P(-1,3)$ $Q(w,7)$ $R(7,4)$

6.* Given the vectors $\mathbf{u} = \langle 2,3 \rangle$ and $\mathbf{v} = \langle -2,-4 \rangle$ find

   (a) $\mathbf{u}+\mathbf{v}$
   (b) $\mathbf{u}-\mathbf{v}$
   (c) $||\mathbf{u}||$ and $||\mathbf{v}||$.
   (d) $||\mathbf{u}+\mathbf{v}||$
   (e) Verify the triangle inequality in this case: $||\mathbf{u}+\mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$
   (f) $-2\mathbf{u}+3\mathbf{v}$

7. Find the unit vector in the direction of v for each of the following.

   (a)* $\mathbf{v} = \langle 3, -4 \rangle$

   (b) $\mathbf{v} = \langle -5, 2 \rangle$

8. Find the unit vector in the direction of $\overrightarrow{PQ}$ where P is the point $(1, 2)$ and Q is the point $(4, 10)$.

9.* Find a vector of length 2 in the direction of $\overrightarrow{PQ}$ where P is the point $(1, 2)$ and Q is the point $(4, 10)$.

10. Here we try to line up the terminal point of a *gun* in such a way that it is pointing in the direction of a given point in space. Assume the initial point (the handle of the gun) is $H(0, 2)$ and the point in space is $(4, 22)$. Make the length of the gun 2. Find the tip of the gun $T(t_1, t_2)$.

11. Express the following vectors as linear combinations of the standard unit vectors.

    (a)* $\mathbf{u} = \langle 2, 3 \rangle$

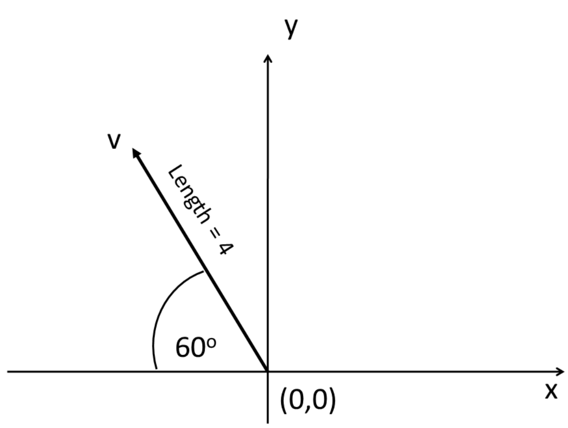
    (b) $\mathbf{v} = \left\langle \sqrt{3}, -7 \right\rangle$

12. Express the requested vector as a linear combination of the standard basis vectors and in component form. Round your answers to two decimal places.

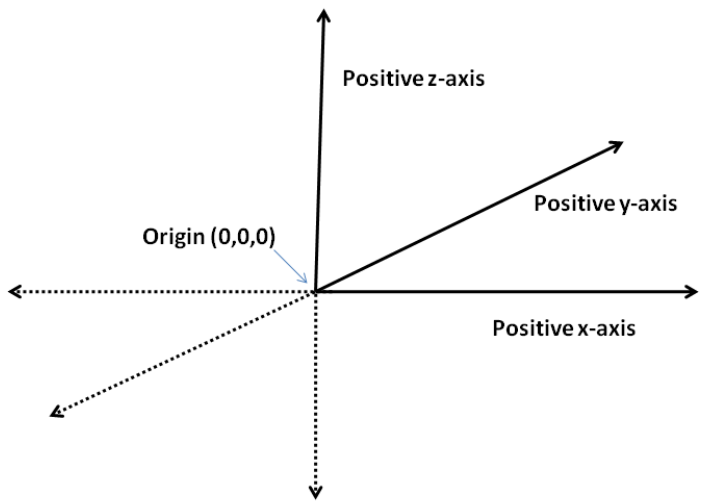
    (a)* The vector is 3 units in length and makes an angle of -120^o with the positive x-axis.

    (b) The vector is 4 units in length and makes an angle of 135^o with the positive x-axis.

13. Give the component form of the vector $\mathbf{v}$ depicted below.
    Give your answer exactly or rounded to 2 decimal places



2.2 Vectors in Space



- The **Three Dimensional Coordinate System** is created by passing a z-axis perpendicular to the usual x and y axes of the cartesian coordinate system. When depicting this system there are two orientations. The one illustrated above represents a *Right-Handed System.* Imagine you are standing at the origin with your arms along the positive x and y axes. The orientation of the system is determined by which hand points along the x-axis. This text will use the right-handed system exclusively but some software programs may use a left-handed system. The default orientation for MATLAB® is also a right-handed system.

- Taken as pairs these axes determine 3 **coordinate planes**; the xy-plane, the xz-plane, and the yz-plane. Visualize these planes in the figure above. These three planes divide 3-space into 8 octants. Unlike 2-space, there is no agreed-upon numbering system for these 8 octants.

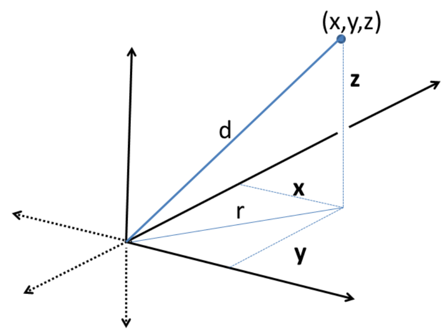
- A **point** in 3 dimensional space (or just space) is determined by an ordered triplet (x, y, z). You start by moving x units along the x-axis, then proceed y units parallel to the y-axis, and finally move z units parallel to the z-axis.

- **Perspective Practice:** Draw your own coordinate axes or use those on the top of this page. Sketch and label the points $P = (2, 3, 4)$, $Q = (-4, 1, 3)$, $R = (-3, -3, 4)$, and $S = (-1, -1, -4)$. Now sketch the directed line segments $\overrightarrow{PQ}$ and $\overrightarrow{RS}$. Try to make them *look real.* What lines are behind others? Using a pencil helps.

Distance

The distance from any point $(x,\ y,\ z)$ to the origin is given by

$$d = \sqrt{x^2 + y^2 + z^2}$$



Mini-Proof: (See Figure)
$r^2 = x^2 + y^2$, and
$d^2 = r^2 + z^2$, so
$d = \sqrt{r^2 + z^2} = \sqrt{x^2 + y^2 + z^2}$.

The **Distance** between points $(x_1,\ y_1,\ z_1)$ and $(x_2,\ y_2,\ z_2)$ is given by

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \tag{2.8}$$

The **midpoint** between points $(x_1,\ y_1,\ z_1)$ and $(x_2,\ y_2,\ z_2)$ is given by

$$\left(\frac{(x_2 + x_1)}{2}, \frac{(y_2 + y_1)}{2}, \frac{(z_2 + z_1)}{2}\right). \tag{2.9}$$

Spheres

Standard equation of a **sphere** with center $(x_0,\ y_0,\ z_0)$ and radius r is

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2. \tag{2.10}$$

- **Example:** Find the standard equation of a sphere with (4, -2, 2) and (0, 5, -2) as endpoints of a diameter.

  Answer: The diameter is given by the distance between the two points.

  $$\text{diameter} = \sqrt{(0-4)^2 + (5-(-2))^2 + (-2-2)^2} = \sqrt{16+49+16} = \sqrt{81} = 9$$

  The radius is half of this, so $r = 4.5$. The center of the sphere is the midpoint of the two points.

  $$\text{center} = \left(\frac{(4+0)}{2},\ \frac{(-2+5)}{2},\ \frac{(2-2)}{2}\right) = \left(2,\ \frac{3}{2},\ 0\right)$$

  Now that we have the radius and center, the equation of the desired sphere is given by equation (2.10),

  $$(x-2)^2 + \left(y - \frac{3}{2}\right)^2 + z^2 = (4.5)^2$$

Vectors in 3-Space

Vectors in 3-space are defined equivalently to those defined previously for 2-space only now we have three components for each vector. Specifically, a **vector in space** is a standardized representation of all equivalent directed line segments in space.

If $\mathbf{v}$ is represented by the line directed line segment from $P = (p_1,\ p_2,\ p_3)$ to $Q = (q_1,\ q_2,\ q_3)$ then the **component form** of $\mathbf{v}$ is given by

$$\mathbf{v} = \langle q_1 - p_1,\ q_2 - p_2,\ q_3 - p_3 \rangle \tag{2.11}$$

Operations and Properties of Vectors in Space

Most of the operations and properties from vectors in the plane extend naturally to space. Review them from the previous lists. Below are a few. Let $\mathbf{u} = \langle u_1,\ u_2,\ u_3 \rangle$, $\mathbf{v} = \langle v_1,\ v_2,\ v_3 \rangle$, and k be a scalar.

Equality: $\mathbf{u} = \mathbf{v}$ only if $u_1 = v_1$, $u_2 = v_2$ and $u_3 = v_3$.

Parallel: Two vectors $\mathbf{u}$ and $\mathbf{v}$ are parallel only if $\mathbf{u} = k\ \mathbf{v}$ for some scalar k.

Length, Norm, or Magnitude: $||\mathbf{v}|| = \sqrt{v_1^2 + v_2^2 + v_3^2}$.

Sum: $\mathbf{u} + \mathbf{v} = \langle u_1 + v_1,\ u_2 + v_2,\ u_3 + v_3 \rangle$

Scalar Multiplication: $k\mathbf{u} = \langle ku_1,\ ku_2,\ ku_3 \rangle$ Note: $||k\mathbf{u}|| = |k|\ ||\mathbf{u}||$

The **unit vector** in the direction of $\mathbf{v}$ is given by $\mathbf{u} = \dfrac{\mathbf{v}}{||\mathbf{v}||}$.

- **Example, Collinear Points:**

  Are the points $P(1,\ -2,\ 3)$, $Q(2,\ 1,\ 0)$, and $R(4,\ 7,\ -4)$ collinear?

  Answer:
  $\overrightarrow{PQ} = \langle 1,\ 3,\ -3 \rangle$ and $\overrightarrow{PR} = \langle 3,\ 9,\ -7 \rangle$. It starts to look like $\overrightarrow{PR} = 3\overrightarrow{PQ}$ but the last term messes this up $\left(\dfrac{3}{1} = \dfrac{9}{3} \neq \dfrac{-7}{-3} \right)$. So, the vectors are not parallel and thus the points are not collinear.

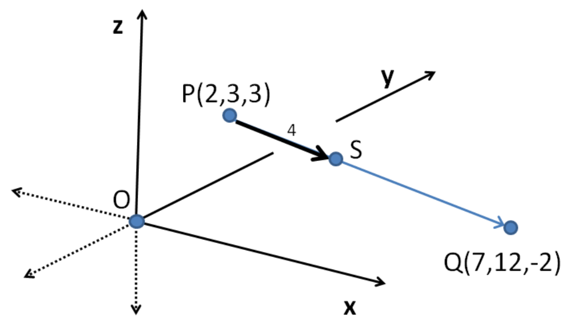
- **Example, Unit Vector:**

  Find the unit vector in the opposite direction of $v = \langle -1,\ 2,\ 3 \rangle$.

  Answer:
  $\mathbf{u} = \dfrac{-1}{||\mathbf{v}||}\mathbf{v} = \dfrac{-1}{\sqrt{14}}\mathbf{v} = \left\langle \dfrac{1}{\sqrt{14}}, \dfrac{-2}{\sqrt{14}}, \dfrac{-3}{\sqrt{14}} \right\rangle$.

- **Example, Point the Arrow**

  Given the points $P(2,3,3)$ and $Q(7,12,-2)$ find the point S such that S is 4 units from P in the direction of $\overrightarrow{PQ}$.



Same strategy as in 2D: First find the unit vector in the direction of $\overrightarrow{PQ}$, then multiply this by 4 to get $\overrightarrow{PS}$. Then add** the vector $\overrightarrow{PS}$ to the point P to get the point S.

- The vector $\overrightarrow{PQ} = \langle 7-2,\ 12-3,\ -2-3\rangle = \langle 5,\ 9,\ -5\rangle$.
- The vector in the direction of $\overrightarrow{PQ}$ of length 4 is $\overrightarrow{PS} = \dfrac{4}{||\overrightarrow{PQ}||}\overrightarrow{PQ} = \dfrac{4}{\sqrt{131}}\langle 5, 9, -5\rangle$.
- Now $S = P + \overrightarrow{PS}$, $S = (2,\ 3,\ 3) + \frac{4}{\sqrt{131}}\ \langle 5,\ 9,\ -5\rangle \approx (3.75,\ 6.15,\ 1.25)$.

** Here again, you can't really add a vector to a point but we allow it here for convenience.

Standard Unit (Basis) Vectors

The standard unit vectors in 3-space are $\mathbf{i} = \langle 1,\ 0,\ 0\rangle$, $\mathbf{j} = \langle 0,\ 1,\ 0\rangle$, and $\mathbf{k} = \langle 0,\ 0,\ 1\rangle$.
Any vector $\mathbf{u} = \langle u_1,\ u_2,\ v_3\rangle$ can be expressed as a linear combination of $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ by

$$\mathbf{u} = u_1\,\mathbf{i} + u_2\,\mathbf{j} + u_3\,\mathbf{k}.$$

- **Examples:**
  - Express $\mathbf{u} = 3\mathbf{i} - 2\mathbf{k}$ in component form.

    Answer: $3\mathbf{i} - 2\mathbf{k} = 3\mathbf{i} + 0\mathbf{j} - 2\mathbf{k} = \langle 3, 0, -2\rangle$.
  - Express $\mathbf{u} = \langle 1, 2, -3\rangle$ as a linear combination of the standard unit vectors.

    Answer: $\langle 1, 2, -3\rangle = \mathbf{i} + 2\mathbf{j} - 3\mathbf{k}$

MATLAB® Demonstration: Plotting line segments, vectors, and points in 2 and 3D.

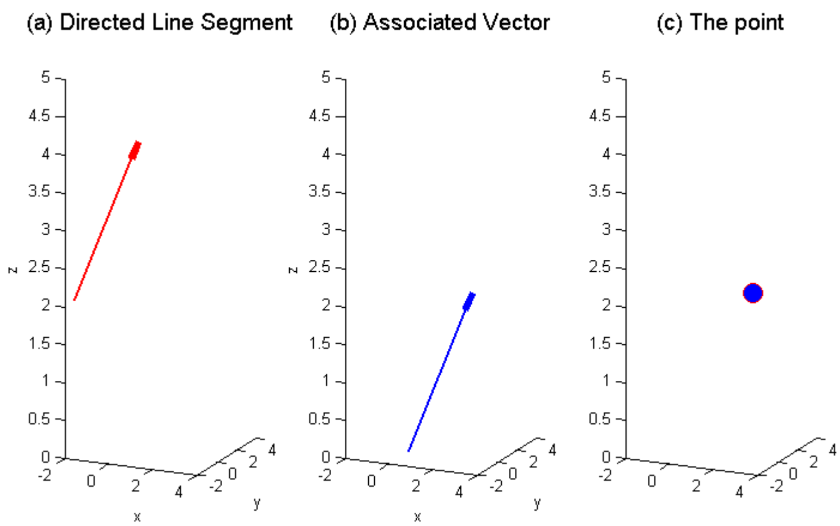
- MATLAB® has many ways to graph 2 and 3 dimensional curves and surfaces that we will investigate in later sections. It does not have a great way of plotting directed line segments and vectors (making the arrow is hard for some reason). The MATLAB® program called `vectarrow.m` is available from the Chapter 2 program repository.

  Usage: `vectarrow(P1,P2,'color')`

- This function file uses the `plot3` command which is good for plotting points and curves in 3D.

  Usage: `plot3([x-values], [y-values], [z-values], ... )`

- The graphs below were generated with these functions. The code is given on the next page.



The code for generating these plots is given on the next page.

Graph (a) depicts the 3-dimensional graph of the directed line segment from P1(-2, -1, 2) to P2(0,1,4) using the command

```
vectarrow([-2, -1, 2], [0, 1, 4], 'red').
```

Graph (b) depicts the associated vector, $\mathbf{u} = \langle 2, 2, 2\rangle$, which has the origin as the initial point. This is created using the same function only now use

```
vectarrow([0,0,0], [2, 2, 2], 'blue').
```

Graph (c) depicts the point at the end of this vector as a large dot in 3-space and is created using the built in `plot3` function by

```
plot3([2], [2], [2], 'ro','markersize', 12, 'markerfacecolor', 'blue').
```

This file (`Plotting_Vectors.m`) is in the Chapter 2 program repository. It generates the three graphs from the previous page. It uses the function file `vectarrow.m` in the Chapter 2 program repository. Make sure both files are in the same folder when you save them.

```
%% This is Plotting_Vectors.m
% Plotting directed line segments, vectors, and points in 3D.
% Uses the function file vectarrow.m
clc; clf; clear;  % clear command window, figure, variables.

%% Plot a Directed Line Segment in Space
subplot(131)  % a 1x3 array of graphs - 1st graph.
P1 = [-2,-1,2]; P2 = [0,1,4];  % P1 = initial point, P2 = terminal point.
vectarrow(P1,P2,'red');  % function file vectarrow.m
axis([-2,4,-2,4,0,5]); % [xmin, xmax, ymin, ymax, zmin, zmax]
view(24,6);  % view(Az,El), found by rotating the graph until you have
             % the desired position, then look in lower left corner.
title('(a) Directed Line Segment','fontsize',14); box on;

%%  Plot a Vector in Space
 subplot(132); % a 1x3 array of graphs - 2nd graph.
 u = P2 - P1;  % The associated vector.
 vectarrow([0,0,0],u,'blue');
 axis([-2,4,-2,4,0,5]); % [xmin, xmax, ymin, ymax, zmin, zmax]
 view(24,6);  % view(Az,El), found by rotating the graph until you have
              % the desired position, then look in lower left corner.
 title('(b) Associated Vector','fontsize',14); box on;

%% Plot a Point in Space
subplot(133);  % a 1x3 array of graphs - 3rd graph.
P = u;   % out point in 3D
plot3(P(1),P(2),P(3),'ro','markersize',12,'markerfacecolor','blue');
axis([-2,4,-2,4,0,5]); % [xmin, xmax, ymin, ymax, zmin, zmax]
view(24,6);  % view(Az,El), found by rotating the graph until you have
             % the desired position, then look in lower left corner.
title('(c) The point','fontsize',14); box on;
```

Chapter 2.2 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. Find the standard equation of the sphere that
   (a)* is centered at (0, -4, 7) with radius = 4.
   (b) has the points (3,0,0) and (0,4,0) as endpoints of a diameter.
   (c)* is centered at (-2,3,4) and is tangent to the xz plane.

2. The initial point and terminal points of a directed line segment are given. (1) Sketch the directed line segment. (2) Write the vector in component form. (3) Sketch the vector with its initial point at the origin.
   (a)* Initial Point: (1,3,-2) Terminal Point: (-2, -7, 8).
   (b) Initial Point: (1,3,0) Terminal Point: (2, 7, 0).

3. Find the vectors $\mathbf{u} = \overrightarrow{PQ}$ and $\mathbf{v} = \overrightarrow{RS}$ for the given points P, Q, R and S. Then (1) determine if $\mathbf{u}$ and $\mathbf{v}$ are equivalent vectors and (2) determine if $\mathbf{u}$ and $\mathbf{v}$ are parallel vectors.
   (a)* $\mathbf{u}$: $P(-2,3,7)$, $Q(-4,-2,9)$ $\mathbf{v}$: $R(2,4,4)$, $S(-2,3,4)$.
   (b) $\mathbf{u}$: $P(0,2,-4)$, $Q(3,0,4)$ $\mathbf{v}$: $R(-1,5,0), S(2,3,8)$.
   (c)* $\mathbf{u}$: $P(2,2,1)$, $Q(5,0,4)$ $\mathbf{v}$: $R(1,3,3), S(-5,7,-3)$.

4. Use vectors to determine whether the points are collinear. Plot the points to confirm your answers.
   (a)* (0, -2, -5), (3, 4, 4), (2, 2, 1)
   (b) (1, 2, 4), (2, 5, 0), (0, 1, 5)

5.* Find the values of x, y, or z needed to make the points P, Q, and R collinear.
   (a)* $P(-1,3,4)$ $Q(2,7,0)$ $R(x,y,7)$
   (b) $P(1,1,-2)$ $Q(2,y,1)$ $R(3,5,z)$

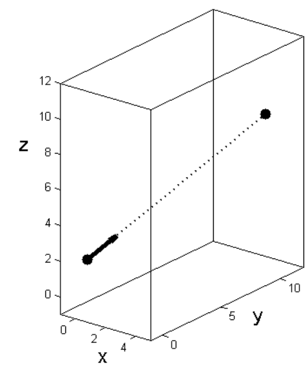
6.* Given the vectors $\mathbf{u} = \langle 2,3,-5 \rangle$ and $\mathbf{v} = \langle -2,-4,0 \rangle$ find
   (a) $\mathbf{u}+\mathbf{v}$
   (b) $\mathbf{u}-\mathbf{v}$
   (c) $||\mathbf{u}||$ and $||\mathbf{v}||$.
   (d) $||\mathbf{u}+\mathbf{v}||$
   (e) Verify the triangle inequality in this case: $||\mathbf{u}+\mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$
   (f) $-2\mathbf{u}+3\mathbf{v}$
   (g) The unit vector in the direction of $\mathbf{v}$.

7. Find the unit vector in the direction of $\overrightarrow{HP}$ where H is the point $(0, 0, 2)$ and P is the point $(4, 10, 8)$.

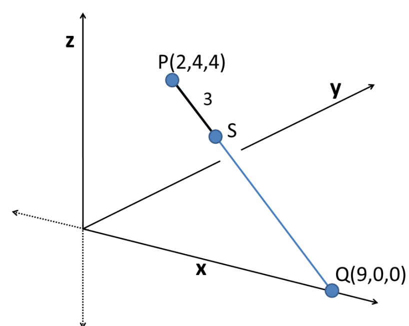
8.* Here we try to line up the tip of an arrow in such a way that it is pointing in the direction of a given point in space. Assume the initial point (the tail of the arrow) is $H(0, 0, 2)$ and the point in space is $(4, 10, 8)$. Make the arrow have length 2. Find the tip of the arrow $T(t_1, t_2, t_3)$.

9. **(MATLAB®)** Write a program that gets the answers to the previous two problems and graphs the point and the arrow in 3D. See `PlottingVectors.m` to help with the plotting commands.

   - Plot the points $H(0, 0, 2)$ and $P(4, 10, 8)$ with a dashed line connecting them.
   - Find the point T (arrow tip) which is 2 units from H in the direction of P. Getting a unit vector in MATLAB® is easily done with the `norm` command: `u = v/norm(v);`, then $T = 2\mathbf{u} + H$.
   - Plot the directed line segment from H to T. You may want to use the `vectarrow.m` program from the textbook's program repository.



10. Using the points P and Q depicted in the figure, find the point S which is three units from the point P in the direction of Q. You should do this by hand and confirm with MATLAB® .

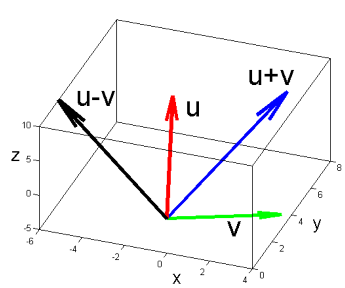


11.* **(MATLAB®)**
Create a program which does the following:

   (a) Plots $\mathbf{u} = \langle -2, 5, 7\rangle$ and $\mathbf{v} = \langle 4, 3, -3\rangle$ in red and green respectively.

   (b) Plots $\mathbf{u} + \mathbf{v}$ in blue and $\mathbf{u} - \mathbf{v}$ in black.

   (c) Make sure you can *see* what is happening with vector addition (tip to tail) and vector subtraction (tip to tip). You may need to rotate the screen to help visualize these.



2.3 The Dot Product

In the previous chapter the **dot product** of two vectors was introduced. It played a critical role in describing how matrix multiplication worked and, as such, how one might solve a system of equations or perform transformations of various types. Here we use the dot product again. In the vector space of $\mathbb{R}^2$ and $\mathbb{R}^3$ (2D and 3D) the dot product is used for many practical purposes such as projecting a vector onto another vector or plane. In my opinion, the dot product is the most important operation in 3D mathematics. Another important operation is the **cross product**, which also has some very practical uses and properties. In this section we investigate the dot product and save the cross product for the next section.

We will start right off with vectors in 3-space. The dot product of $\mathbf{u}$ and $\mathbf{v}$ is defined by

$$\mathbf{u}\cdot\mathbf{v} = \langle u_1,\ u_2,\ u_3\rangle \cdot \langle v_1,\ v_2,\ v_3\rangle = u_1v_1 + u_2v_2 + u_3v_3 \tag{2.12}$$

which should be familiar to you from the previous chapter. The dot product described here satisfies the same properties as the dot product described in the previous chapter. For example: $\mathbf{u}\cdot(\mathbf{v}+\mathbf{w}) = \mathbf{u}\cdot\mathbf{v} + \mathbf{u}\cdot\mathbf{w}$. While we are reviewing stuff, it is worth noting that the dot product can help us get the length of a vector by the equation

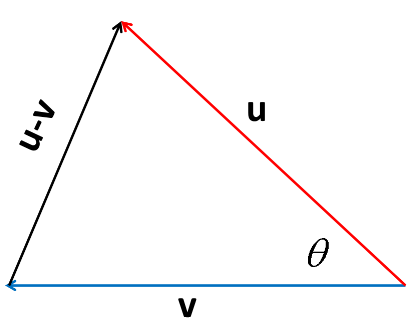
$$||\mathbf{u}||^2 = \mathbf{u}\cdot\mathbf{u} \tag{2.13}$$

because $||\mathbf{u}||^2 = \left(\sqrt{u_1^2+u_2^2+u_3^2}\right)^2 = u_1^2+u_2^2+u_3^2 = \mathbf{u}\cdot\mathbf{u}$.

Before continuing, it is worth noting that two distinct vectors in space define a plane in that space. In that plane, there is an angle created by the two vectors $\mathbf{u}$ and $\mathbf{v}$. We will call that angle θ and assume that $0 \le \theta \le \pi$ $(0 \le \theta \le 180^o)$. What makes the dot product so special is the equivalent definition which can be derived using the law of cosines:

$$\mathbf{u}\cdot\mathbf{v} = ||\mathbf{u}||\ ||\mathbf{v}||\ \cos(\theta). \tag{2.14}$$

♣ **Mini-Proof:**



By the *Law of Cosines*

$$\begin{aligned}
||\mathbf{u}||^2 + ||\mathbf{v}||^2 - 2\ ||\mathbf{u}||\ ||\mathbf{v}||\ \cos\theta &= ||\mathbf{u}-\mathbf{v}||^2 \\
&= (\mathbf{u}-\mathbf{v})\cdot(\mathbf{u}-\mathbf{v}) \\
&= \mathbf{u}\cdot\mathbf{u} - 2\ \mathbf{u}\cdot\mathbf{v} + \mathbf{v}\cdot\mathbf{v} \\
&= ||\mathbf{u}||^2 - 2\ \mathbf{u}\cdot\mathbf{v} + ||\mathbf{v}||^2
\end{aligned}$$

and subtracting $||\mathbf{u}||^2 + ||\mathbf{v}||^2$ from both sides

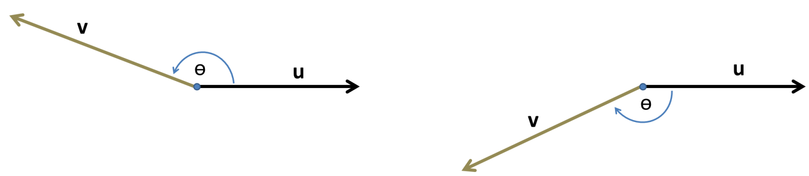
$$\begin{aligned}
-2\ ||\mathbf{u}||\ ||\mathbf{v}||\ \cos\theta &= -2\ \mathbf{u}\cdot\mathbf{v} \\
||\mathbf{u}||\ ||\mathbf{v}||\ \cos\theta &= \mathbf{u}\cdot\mathbf{v} \qquad ♣
\end{aligned}$$

The Angle Between Vectors

Equation (2.14) can be used to obtain the angle between two vectors.

$$\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} \quad \text{or} \quad \theta = \arccos\left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||}\right) \qquad 0 \le \theta \le \pi \tag{2.15}$$

Note: The arccosine function is denoted arccos in text. On calculators it is often denoted $\cos^{-1}$ (cosine inverse). In software, it is usually called with a function like `acos`. This function always returns an angle between 0 and π radians (or 0 and 180°) as illustrated below.



- **Examples:** Determine the cosine of the angle and the angle between the given vectors.

  1. $\mathbf{u} = \langle 1,\ -3\rangle$ and $\mathbf{v} = \langle 1,\ 2\rangle$.

     Answer: Using equation (2.15), $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \frac{-5}{\sqrt{10}\,\sqrt{5}} = \frac{-5}{\sqrt{50}} = \frac{-5}{5\sqrt{2}} = \frac{-1}{\sqrt{2}} = \frac{-\sqrt{2}}{2}$, and $\theta = \arccos\left(\frac{-\sqrt{2}}{2}\right) = \frac{3\pi}{4}$ radians or 135° .

  2. $\mathbf{u} = \langle 1, 2, 3\rangle$ and $\mathbf{v} = \langle -1, 3, 5\rangle$.

     Answer: Using equation (2.15), $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \frac{20}{\sqrt{14}\,\sqrt{35}}$.
     Using software, $\theta = \arccos\left(\frac{20}{\sqrt{14}\,\sqrt{35}}\right) \approx 0.443$ radians or 25.4^o.

Orthogonal Vectors: Two vectors are called orthogonal if the angle between them is $\pi/2$ radians (90°). The convention is to use the term orthogonal as opposed to perpendicular when discussing vectors. Since $\cos\left(\frac{\pi}{2}\right) = 0$, equation (2.15) implies that the dot product of two orthogonal vectors is always zero. This acts as an additional definition for orthogonal vectors.

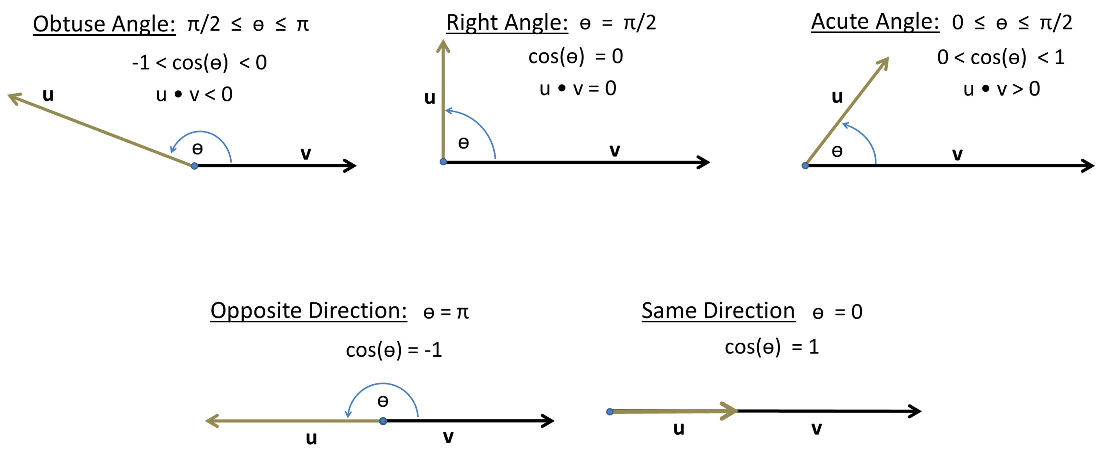
Two vectors $\mathbf{u}$ and $\mathbf{v}$ are orthogonal if $\mathbf{u}\cdot\mathbf{v} = 0$.

- **Example:** Prove that the vectors $\mathbf{u} = \langle 1,\ -2,\ 3\rangle$ and $\mathbf{v} = \langle 2,\ -2,\ -2\rangle$ are orthogonal.

  Answer: $\mathbf{u}\cdot\mathbf{v} = 2 + 4 - 6 = 0$ and the vectors are orthogonal.

More About Angles Without Any Trig

In addition to getting the angle between vectors, equation (2.15) allows us to learn a lot about the angle between two vectors without any trigonometry. The following sequence of graphs depicts this information.



- **Example 1:** Using only the dot product, determine if the angle between $\mathbf{u} = \langle 1,\ -11\rangle$ and $\mathbf{v} = \langle 3,\ 5\rangle$ is an acute, obtuse, or right angle.

  Answer: $\mathbf{u}\cdot\mathbf{v} = 3 - 55 = -52$. Since this is less than zero, the angle is obtuse.

- **Example 2:** Determine if the points $A(1,\ 2,\ 5)$, $B(3,\ -4,\ 13)$, and $C(0,\ 5,\ 1)$ are collinear.

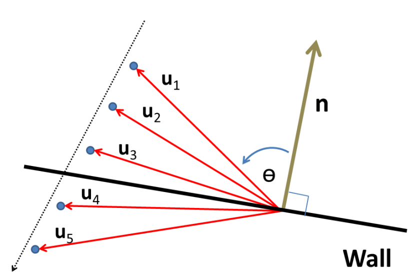
  Answer: Let $\mathbf{u} = \overrightarrow{AB} = \langle 2,\ -6,\ 8\rangle$ and $\mathbf{v} = \overrightarrow{AC} = \langle -1,\ 3,\ -4\rangle$ and θ be the angle between $\mathbf{u}$ and $\mathbf{v}$.

  $$\cos\theta = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||} = \frac{-52}{\sqrt{104}\,\sqrt{26}} = \frac{-52}{2\sqrt{26}\,\sqrt{26}} = \frac{-52}{52} = -1$$

  Since $\cos(\theta) = -1$, the vectors are in opposite directions and the points are collinear.

Application - Collision Detection

Suppose there is a wall with an orthogonal vector $\mathbf{n}$ and a ball moving towards the wall following $\mathbf{u}_1,\ \mathbf{u}_2,\ \ldots,\ \mathbf{u}_5$. Between $\mathbf{u}_3$ and $\mathbf{u}_4$, the angle (θ) changes from acute to obtuse. That means the sign of $\mathbf{n}\cdot\mathbf{u}_i$ changes when this happens. As such, if you want to see if or when the ball hits the wall you only need to determine if or when the sign of this dot product changes. Easy!



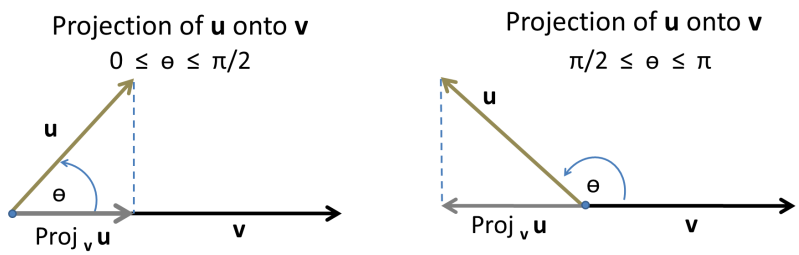
Projections

The projection of $\mathbf{u}$ onto $\mathbf{v}$, denoted $\text{Proj}_{\mathbf{v}}\mathbf{u}$, is given by

$$\text{Proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v}. \tag{2.16}$$

This will prove to be very useful. We are able to do projections without any trigonometry!

♣ **Mini-Proof:** This must be done in two separate situations: $0 \le \theta \le \pi/2$ and $\pi/2 \le \theta < \pi$. If $\theta = \pi/2$ (90^o) then the projection equals the zero vector.



$$\text{Proj}_{\mathbf{v}}\mathbf{u} = ||\mathbf{u}||\cos(\theta)\ \frac{\mathbf{v}}{||\mathbf{v}||} = ||\mathbf{u}||\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||}\ \frac{\mathbf{v}}{||\mathbf{v}||} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v} \qquad ♣$$

- **Examples:**

1. Let $\mathbf{u} = \langle -2,\ 2\rangle$ and $\mathbf{v} = \langle 0,\ 3\rangle$ find $\text{Proj}_{\mathbf{v}}\mathbf{u}$, $\text{Proj}_{\mathbf{u}}\mathbf{v}$, and $\text{Proj}_{(-\mathbf{u})}\mathbf{v}$.

   $\text{Proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v} = \left(\frac{6}{9}\right)\ \langle 0,\ 3\rangle = \langle 0,\ 2\rangle$. Notice: $\text{Proj}_{\mathbf{v}}\mathbf{u}$ is parallel to $\mathbf{v}$.

   $\text{Proj}_{\mathbf{u}}\mathbf{v} = \left(\frac{\mathbf{v}\cdot\mathbf{u}}{||\mathbf{u}||^2}\right)\ \mathbf{u} = \left(\frac{6}{8}\right)\ \langle -2,\ 2\rangle = \langle -1.5,\ 1.5\rangle$. Notice: $\text{Proj}_{\mathbf{u}}\mathbf{v}$ is parallel to $\mathbf{u}$.

   $\text{Proj}_{(-\mathbf{u})}\mathbf{v} = \left(\frac{\mathbf{v}\cdot(-\mathbf{u})}{||-\mathbf{u}||^2}\right)\ (-\mathbf{u}) = \left(\frac{-6}{8}\right)\ \langle 2,\ -2\rangle = \langle -1.5,\ 1.5\rangle$. Notice: $\text{Proj}_{(-\mathbf{u})}\mathbf{v} = \text{Proj}_{\mathbf{u}}\mathbf{v}$

2. Let $\mathbf{u} = \langle -2,\ 2,\ 1\rangle$ and $\mathbf{v} = \langle 1,\ 0,\ 3\rangle$ find $\text{Proj}_{\mathbf{v}}\mathbf{u}$, $\text{Proj}_{\mathbf{u}}\mathbf{v}$, and $\text{Proj}_{\mathbf{u}}(-\mathbf{v})$.

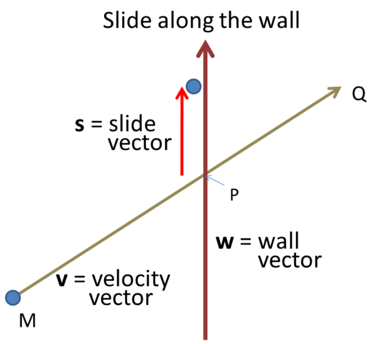
   $\text{Proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v} = \left(\frac{1}{10}\right)\ \langle 1,\ 0,\ 3\rangle = \langle 0.1,\ 0,\ 0.3\rangle$. Notice: $\text{Proj}_{\mathbf{v}}\mathbf{u}$ is parallel to $\mathbf{v}$.

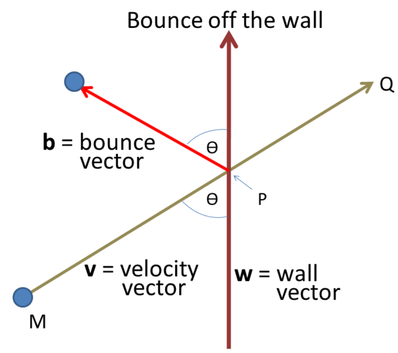
   $\text{Proj}_{\mathbf{u}}\mathbf{v} = \left(\frac{\mathbf{v}\cdot\mathbf{u}}{||\mathbf{u}||^2}\right)\ \mathbf{u} = \left(\frac{1}{9}\right)\ \langle -2,\ 2,\ 1\rangle = \left\langle \frac{-2}{9},\ \frac{2}{9},\ \frac{1}{9}\right\rangle$. Notice: $\text{Proj}_{\mathbf{u}}\mathbf{v}$ is parallel to $\mathbf{u}$.

   $\text{Proj}_{\mathbf{u}}(-\mathbf{v}) = \left(\frac{-\mathbf{v}\cdot\mathbf{u}}{||\mathbf{u}||^2}\right)\ \mathbf{u} = \left(\frac{-1}{9}\right)\ \langle -2,\ 2,\ 1\rangle = \left\langle \frac{2}{9},\ \frac{-2}{9},\ \frac{-1}{9}\right\rangle$. Notice: $\text{Proj}_{\mathbf{u}}(-\mathbf{v}) = -\text{Proj}_{\mathbf{u}}\mathbf{v}$.

Applications - Slide Response and Bounce Response

As an object moves through 2 or 3 space it will run into other objects. Let's consider the case where an object hits a wall. There are two standard responses to hitting a wall. The object can *slide* along the wall in a way indicative of the way how it hit the wall. On the other hand, if we are talking about a ball or some projectile, it may bounce off the wall at an appropriate angle (angle of incidence = angle of reflection). For now we stick with 2D because in 3D we will look at collisions with planes and we haven't studied these yet. Let M represent the initial point of the object. Let the vector $\mathbf{v} = \overrightarrow{MQ}$ represent the velocity vector which tells us the direction and distance of motion. At point P, the object hits the wall defined by the vector $\mathbf{w}$.





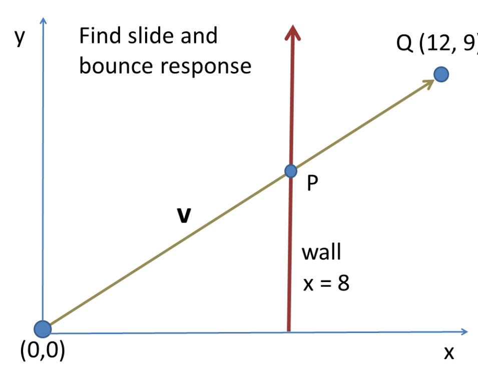
slide vector: $\mathbf{s} = \text{Proj}_{\mathbf{w}}\overrightarrow{PQ}$ (2.17)

bounce vector: $\mathbf{b} = 2\mathbf{s} - \overrightarrow{PQ}$ (2.18)

slide point = $P + \mathbf{s}$

bounce point = $P + \mathbf{b}$

• **Example:**. Determine the final location of the guy (or ball) for a slide and a bounce if $M = (0,\ 0)$, $\mathbf{v} = \langle 12,\ 9\rangle$, and a wall is at $x = 8$.



Answers: First you need P at the intersection of $\mathbf{v}$ and $x = 8$. The line containing $\mathbf{v}$ is $y = 9/12\,x$. When $x = 8$, $y = 6$. So $P = (8,\ 6)$.

$\mathbf{w} = \langle 0,\ k\rangle$ for any $k > 0$.

$\overrightarrow{PQ} = Q(12,\ 9) - P(8,\ 6) = \langle 4,\ 3\rangle$

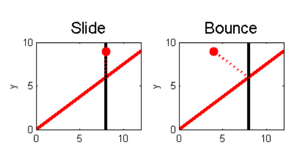
Slide: $\mathbf{s} = \text{Proj}_{\mathbf{w}}\overrightarrow{PQ} = \left(\dfrac{\mathbf{w}\cdot\overrightarrow{PQ}}{||\mathbf{w}||^2}\right)\mathbf{w} = \dfrac{3k}{k^2}\,\langle 0, k\rangle = \langle 0, 3\rangle$, and the final point is $(8, 9)$.

Bounce: $\mathbf{b} = 2\mathbf{s} - \overrightarrow{PQ} = 2\,\langle 0, 3\rangle - \langle 4, 3\rangle = \langle -4, 3\rangle$, and the final point is $(4, 9)$.

MATLAB® Demonstration: Here we use MATLAB® to perform the last example. `ProjUV.m` and `Slide_Bounce.m` are available from the Chapter 2 program repository.

```
function [w] = ProjUV(u,v)
% Project the vector u in the direction of v
% Vectors u and v must be the same length.
% This returns w with the same dimensions as v.
if length(u) == length(v)
    w = dot(u,v)/(norm(v))^2 * v;
else
    error('vectors must be the same length')
    abort;
end
```

```
%% Slide_Bounce.m
% Calculating projections with ProjUV function and plotting in 2D.
% Demonstrating a slide and bounce response
clc; clf; clear;  % clear command window, figure, and all variables

M = [0 0];   % origin
P = [8 6];   % collision point (you have to determine this)
Q = [12 9]; % terminal point of v
w1 = [8 0];  % bottom of the wall
w2 = [8 10]; % top of the wall
PQ = Q - P;  % vector PQ
w = w2 - w1;  % wall vector
s = ProjUV(PQ,w)  % slide vector = Projection of PQ onto w.
b = 2*s - PQ      % bounce vector
SlidePoint = P + s  % endpoint of slide
BouncePoint = P + b % endpoint of bounce

subplot(121)
plot([M(1),Q(1)],[M(2),Q(2)],'r-','linewidth',3); hold on;
plot([w1(1),w2(1)],[w1(2),w2(2)], 'k-','linewidth',3);
plot([SlidePoint(1)],[SlidePoint(2)], 'r.','markersize',30);
plot([P(1),SlidePoint(1)],[P(2),SlidePoint(2)], 'r:','linewidth',3);
xlabel('x'); ylabel('y'); title('Slide','fontsize',16);
axis equal; axis([0,12,0,10]);

subplot(122)
plot([M(1),Q(1)],[M(2),Q(2)],'r-','linewidth',3); hold on;
plot([w1(1),w2(1)],[w1(2),w2(2)], 'k-','linewidth',3);
plot([BouncePoint(1)],[BouncePoint(2)], 'r.','markersize',30);
plot([P(1),BouncePoint(1)],[P(2),BouncePoint(2)], 'r:','linewidth',3);
xlabel('x'); ylabel('y'); title('Bounce','fontsize',16);
axis equal; axis([0,12,0,10]);
```



Chapter 2.3 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. For the following vectors, find a) $\mathbf{u} \cdot \mathbf{v}$, b) $\mathbf{u} \cdot \mathbf{u}$, c) $||\mathbf{u}||^2$, d) $2(\mathbf{u} \cdot \mathbf{v})\, \mathbf{v}$

   (a)* $\mathbf{u} = \langle 3, 4\rangle$ and $\mathbf{v} = \langle 2, -3\rangle$

   (b) $\mathbf{u} = \langle 2, -3, 4\rangle$ and $\mathbf{v} = \langle 1, 0, -1\rangle$

2. Find $\mathbf{u} \cdot \mathbf{v}$

   (a)* $||\mathbf{u}|| = 8$, $||\mathbf{v}|| = 3$, and the angle between $\mathbf{u}$ and $\mathbf{v}$ is 35^o.

   (b) $||\mathbf{u}|| = 2$, $||\mathbf{v}|| = 6$, and the angle between $\mathbf{u}$ and $\mathbf{v}$ is $\frac{\pi}{2}$.

3. Find the angle θ between the vectors.

   (a)* $\mathbf{u} = \langle 1, 1\rangle$ and $\mathbf{v} = \langle 2, -2\rangle$

   (b) $\mathbf{u} = \langle 1, 1, 1\rangle$ and $\mathbf{v} = \langle 2, 1, -1\rangle$

4. Find the cosine of the angle between the given vectors. Determine whether $\mathbf{u}$ and $\mathbf{v}$ are orthogonal, parallel (same direction or opposite), or neither. If neither, determine whether the angle between them is acute or obtuse.

   (a)* $\mathbf{u} = \langle 4, 0\rangle$ and $\mathbf{v} = \langle 1, 1\rangle$

   (b) $\mathbf{u} = \langle 2, 18\rangle$ and $\mathbf{v} = \left\langle \frac{3}{2}, \frac{-1}{6} \right\rangle$

   (c)* $\mathbf{u} = \langle 2, -3, 1\rangle$ and $\mathbf{v} = \langle -1, 1, -1\rangle$

   (d) $\mathbf{u} = \langle \cos(\theta), \sin(\theta), -1\rangle$ and $\mathbf{v} = \langle \sin(\theta), -\cos(\theta), 0\rangle$

5. Determine the value of w needed to make $\mathbf{u}$ and $\mathbf{v}$ orthogonal.

   (a)* $\mathbf{u} = \langle 3, -2, 1\rangle$ is orthogonal to $\mathbf{v} = \langle 1, 2, w\rangle$.

   (b) $\mathbf{u} = \langle 3, w, 1\rangle$ is orthogonal to $\mathbf{v} = \langle w, 5, 4\rangle$.

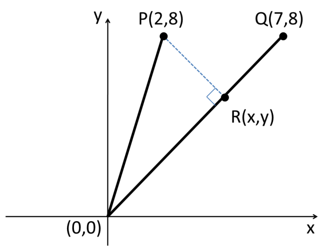
6. Find $\text{Proj}_{\mathbf{v}}\mathbf{u}$ for the following vectors.

   (a)* $\mathbf{u} = \langle 4, 0\rangle$ and $\mathbf{v} = \langle 1, 1\rangle$

   (b) $\mathbf{u} = \langle 2, 18\rangle$ and $\mathbf{v} = \left\langle \frac{3}{2}, \frac{-1}{6} \right\rangle$

   (c)* $\mathbf{u} = \langle 2, -3, 1\rangle$ and $\mathbf{v} = \langle -1, 1, -1\rangle$

7. Determine the point R from the figure. **Give your answer exactly or round to two decimal places**.



8.* Determine which of the following are well-defined for nonzero vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ of equal length.

(a) $\mathbf{u}\cdot(\mathbf{v}+\mathbf{w})$

(b) $\mathbf{u}\cdot\mathbf{v}+\mathbf{w}$

(c) $(\mathbf{u}\cdot\mathbf{v})\,\mathbf{w}$

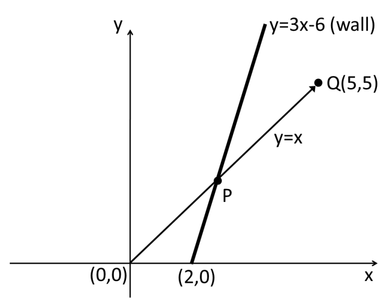
(d) $(\mathbf{u}\cdot\mathbf{v})\cdot\mathbf{w}$

(e) $||\mathbf{u}||(\mathbf{v}+\mathbf{w})$

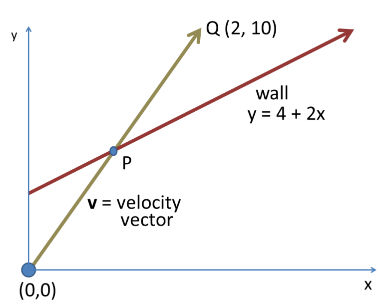
(f) $\dfrac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{w}}$

9.* (**MATLAB®**): Given $\mathbf{u}=\langle 3,-1,2\rangle$ and $\mathbf{v}=\langle 2,2,3\rangle$. Use MATLAB® to plot both vectors in 3D. On the same graph plot the the projection of $\mathbf{u}$ onto $\mathbf{v}$. Title the graph to distinguish which vector is which. Finally, use the `acos` (MATLAB® 's arccosine function) to determine the angle between $\mathbf{u}$ and $\mathbf{v}$ in **degrees**.

10.* A ball starts at (0,0) and wants to follow the line $y=x$ to the point (5,5). It hits a wall defined by $y=3x-6$ at the point P. Find the final point of a slide response and a bounce response from this collision. Check your answers with MATLAB® .



11. (**MATLAB®**): A ball starts at (0,0) and follows the velocity vector shown in the figure. It eventually hits the wall at point P. Use MATLAB® to demonstrate a slide response and a bounce response to the collision. Return the endpoint of each type of response. Use `axis equal;` to make sure these responses *look* right. You can start by editing `Slide_Bounce.m` from the program repository



12. Use the definition of the $\text{Proj}_{\mathbf{v}}\mathbf{u}$ to prove the following identities.

(a)* $\text{Proj}_{(-\mathbf{v})}\mathbf{u}=\text{Proj}_{\mathbf{v}}\mathbf{u}$

(b) $\text{Proj}_{\mathbf{v}}(-\mathbf{u})=-\text{Proj}_{\mathbf{v}}\mathbf{u}$.

2.4 The Cross Product

The cross product is very different from the dot product. First, the cross product is used almost exclusively for vectors in $\mathbb{R}^3$. Second, the dot product returns a scalar while the cross product returns a vector. The important feature of the cross product is that the resulting vector is orthogonal (perpendicular) to both of the original vectors. We start with the formal definition of the cross-product.
If $\mathbf{u} = \langle u_1,\ u_2,\ u_3\rangle$ and $\mathbf{v} = \langle v_1,\ v_2,\ v_3\rangle$ then the **cross product** of $\mathbf{u}$ and $\mathbf{v}$ is

$$\mathbf{u}\times\mathbf{v} = \langle (u_2v_3 - u_3v_2),\ -(u_1v_3 - u_3v_1),\ (u_1v_2 - u_2v_1)\rangle \tag{2.19}$$

The above definition is difficult (or impossible) to remember. There is another way of describing this formula when the vectors are written in terms of the standard unit vectors and the cross product is described by the determinant of a special matrix.

If $\mathbf{u} = u_1\,\mathbf{i} + u_2\,\mathbf{j} + u_3\,\mathbf{k}$ and $\mathbf{v} = v_1\,\mathbf{i} + v_2\,\mathbf{j} + v_3\,\mathbf{k}$, then

$$\mathbf{u}\times\mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix}$$

or

$$\mathbf{u}\times\mathbf{v} = (u_2v_3 - u_3v_2)\,\mathbf{i} - (u_1v_3 - u_3v_1)\,\mathbf{j} + (u_1v_2 - u_2v_1)\,\mathbf{k} \tag{2.20}$$

Algebraic Properties If $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are vectors and c is a scalar

$\mathbf{u}\times\mathbf{v} = -(\mathbf{v}\times\mathbf{u})$

$\mathbf{u}\times(\mathbf{v}+\mathbf{w}) = \mathbf{u}\times\mathbf{v} + \mathbf{u}\times\mathbf{w}$

$c\,(\mathbf{u}\times\mathbf{v}) = c\,\mathbf{u}\times\mathbf{v} = \mathbf{u}\times c\,\mathbf{v}$

$\mathbf{u}\times\mathbf{0} = \mathbf{0}\times\mathbf{u} = \mathbf{0}$

$\mathbf{u}\times\mathbf{u} = \mathbf{0}$

$\mathbf{u}\cdot(\mathbf{v}\times\mathbf{w}) = (\mathbf{u}\times\mathbf{v})\cdot\mathbf{w}$

Geometric Properties If $\mathbf{u}$ and $\mathbf{v}$ are vectors, θ is the angle between them, and c is a scalar:

$\mathbf{u}\times\mathbf{v}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.

$||\mathbf{u}\times\mathbf{v}|| = ||\mathbf{u}||\ ||\mathbf{v}||\ \sin\theta$

$||c\mathbf{u}\times\mathbf{v}|| = ||\mathbf{u}\times c\mathbf{v}|| = |c|\ ||\mathbf{u}\times\mathbf{v}||$

$\mathbf{u}\times\mathbf{v} = \mathbf{0}$ if and only if $\mathbf{u}$ and $\mathbf{v}$ are scalar multiples of each other (parallel).

$||\mathbf{u}\times\mathbf{v}||$ = the area of the parallelogram having $\mathbf{u}$ and $\mathbf{v}$ as adjacent sides.

$|\mathbf{w}\cdot(\mathbf{u}\times\mathbf{v})|$ = the volume of the parallelepiped spanned by $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$.

- **Example:** Let $\mathbf{u} = \langle 0,\ 1,\ 5 \rangle$ and $\mathbf{v} = \langle -1,\ 0,\ -2 \rangle$, find $\mathbf{u} \times \mathbf{v}$ and $\mathbf{v} \times \mathbf{u}$.
  Then verify (a) that $\mathbf{u} \times \mathbf{v} = -\mathbf{v} \times \mathbf{u}$, (b) that $\mathbf{u} \times \mathbf{v}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$, and (c) that $\mathbf{v} \times \mathbf{u}$ is orthogonal to $\mathbf{u}$ and $\mathbf{v}$.

  Answers:

  Let $\mathbf{w}_1 = \mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 1 & 5 \\ -1 & 0 & -2 \end{vmatrix} = -2\mathbf{i} - 5\mathbf{j} + 1\mathbf{k} = \langle -2, -5, 1 \rangle$.

  Let $\mathbf{w}_2 = \mathbf{v} \times \mathbf{u} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -1 & 0 & -2 \\ 0 & 1 & 5 \end{vmatrix} = 2\mathbf{i} + 5\mathbf{j} - 1\mathbf{k} = \langle 2, 5, -1 \rangle$.

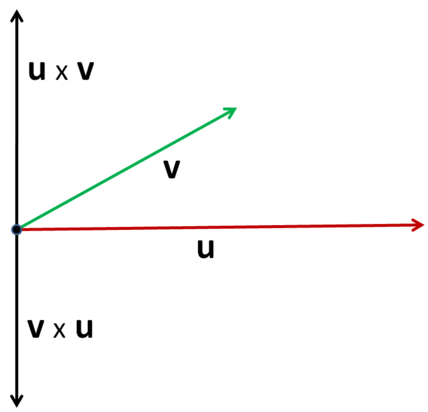
  (a) Notice from above that $\mathbf{u} \times \mathbf{v} = -\mathbf{v} \times \mathbf{u}$.

  (b) Recall, two vectors are orthogonal if their dot product is zero. Notice $\mathbf{w}_1 \cdot \mathbf{u} = 0$ and $\mathbf{w}_1 \cdot \mathbf{v} = 0$, therefore $\mathbf{u} \times \mathbf{v}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.

  (c) Likewise, $\mathbf{w}_2 \cdot \mathbf{u} = 0$ and $\mathbf{w}_2 \cdot \mathbf{v} = 0$, therefore $\mathbf{v} \times \mathbf{u}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.

The Right-Hand Rule

We know that $\mathbf{u} \times \mathbf{v}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$. Consider the plane containing $\mathbf{u}$ and $\mathbf{v}$. There are two possible vectors that are orthogonal to both $\mathbf{u}$ and $\mathbf{v}$. They point in opposite directions. Which one is $\mathbf{u} \times \mathbf{v}$? We use the right-hand rule to make this determination. Hold out your right hand and curl your fingers from $\mathbf{u}$ to $\mathbf{v}$ via the smaller angle. Now, your thumb is pointing in the direction of $\mathbf{u} \times \mathbf{v}$.

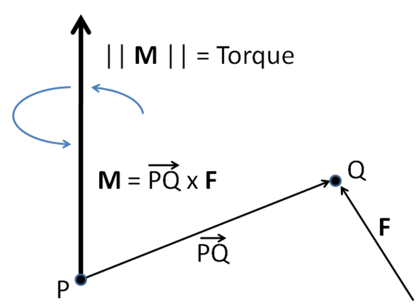


Application in Physics - Torque

The **moment M** of a force **F** about a point P applied at the point Q is given by

$$\mathbf{M} = \overrightarrow{PQ} \times \mathbf{F}.$$

This is also called the **torque vector** and $||\mathbf{M}||$ is called the **torque** which is a measure of the tendency for the vector $\overrightarrow{PQ}$ to rotate counterclockwise (using the right hand rule) about the axis directed along the vector **M**.



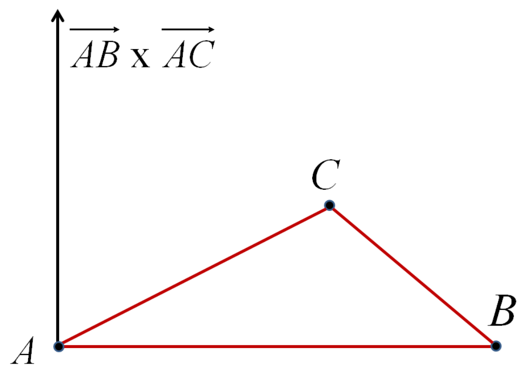
Notice, if $\overrightarrow{PQ}$ is longer (longer wrench) or the force **F** is increased, then the torque is increased.

Application in Graphics: Back-Face Culling, Clockwise or Counter-Clockwise

When rendering images on a computer, you may only want to render the surfaces facing the camera. The vertices of objects can be ordered in such a way that only those appearing (from the camera's perspective) in a clockwise order will be rendered. So the question becomes; How can we take a set of points and a given camera position and determine whether or not the ordered points *appear* in a clockwise or counter-clockwise fashion from the camera's perspective? This process is called back-face culling.

Getting Started:

- Let A, B, and C represent the vertices in question
- $\overrightarrow{AB} \times \overrightarrow{AC}$ produces a vector orthogonal to both $\overrightarrow{AB}$ and $\overrightarrow{AC}$.
- Using the *right-hand-rule*, the vertices A, B, and C appear in a *counter-clockwise* fashion from the perspective of any point on $\overrightarrow{AB} \times \overrightarrow{AC}$



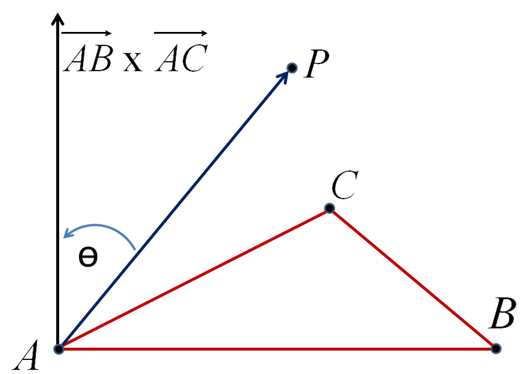
Now we introduce a point P representing the camera position. Recall from the previous section

- If $\mathbf{u} \cdot \mathbf{v} > 0$ the angle between $\mathbf{u}$ and $\mathbf{v}$ is acute ($< 90^o$).
- If $\mathbf{u} \cdot \mathbf{v} < 0$ the angle between $\mathbf{u}$ and $\mathbf{v}$ is obtuse ($> 90^0$).

Let θ be the angle between $\overrightarrow{AP}$ and $\overrightarrow{AB} \times \overrightarrow{AC}$.

If $\overrightarrow{AP} \cdot \left(\overrightarrow{AB} \times \overrightarrow{AC}\right) > 0$ then $\theta < 90^o$

If $\overrightarrow{AP} \cdot \left(\overrightarrow{AB} \times \overrightarrow{AC}\right) < 0$ then $\theta > 90^o$



With respect to P, the orientation of A, B, and C is

$$\begin{array}{lcl} \text{counter-clockwise} & \iff & \overrightarrow{AP} \cdot \left(\overrightarrow{AB} \times \overrightarrow{AC}\right) > 0 \\ \text{clockwise} & \iff & \overrightarrow{AP} \cdot \left(\overrightarrow{AB} \times \overrightarrow{AC}\right) < 0 \end{array}$$

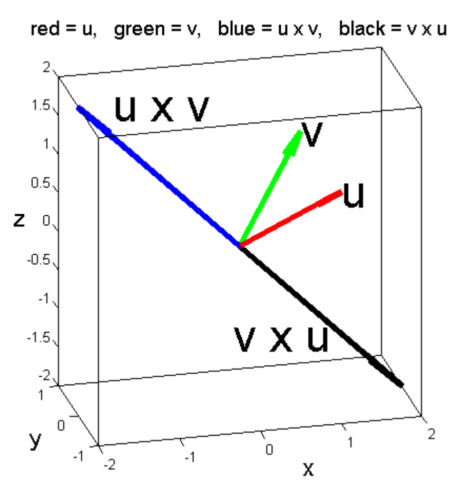
MATLAB® Demonstration for Cross Products

MATLAB® has a built-in cross product function called `cross`. Here again, we use the `vectarrow` function to graph the vectors. This file (`Plotting_Cross_Products.m`) is in the Chapter 2 program repository.

```
%% Plotting_Cross_Products.m
clc; clf; clear;  %% clear command window, figure, and variables

%% Start
u = [1, -1, 1]; v = [1, 1, 1]; Ovec = [0, 0, 0];
w1 = cross(u,v);  w2 = cross(v,u);

%% Plot 2 vectors and their cross product using  vectarrow.m.
vectarrow(Ovec,u,'red');  hold on;
vectarrow(Ovec,v,'green'); vectarrow(Ovec,w1,'blue'); ...
    vectarrow(Ovec,w2,'black');
title('red = u,   green = v,   blue = u x v,   black = v x ...
    u','fontsize',14)
box on; view(-14,22); axis equal;

%% Labeling Things in the Graph:
xlabel('x','fontsize',16);ylabel('y','fontsize',16);
zlabel('z','fontsize',16);set(get(gca,'ZLabel'),'Rotation',0.0)
text(u(1),u(2),u(3),'u','fontsize',32);
text(v(1),v(2),v(3),'v','fontsize',32);
text(w1(1),w1(2),w1(3),'u x v','fontsize',32);
text(w2(1),w2(2),w2(3),'v x u','fontsize',32);
```

Some Notes About This Code

- Syntax: `cross(u,v)` $= \mathbf{u} \times \mathbf{v}$
  Order matters.
- The `vectarrow.m` file is in the Chapter 2 program repository.
- The arguments `Az` and `El` in `view(Az,El)` are best found by first rotating the graph with the rotation button and then choosing the angles (lower left of graph during rotation) that look best.
- The command, `axis equal;` ensures the cross products look perpendicular to both vectors under all viewing positions.



Chapter 2.4 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1.* Suppose $\mathbf{u} \times \mathbf{v} = \mathbf{w}$. What can be said about the following?

   (a) $\mathbf{v} \times \mathbf{u}$

   (b) $-\mathbf{u} \times \mathbf{v}$

   (c) $||2\ \mathbf{u} \times 2\ \mathbf{v}||$

   (d) $\mathbf{w} \cdot \mathbf{u}$

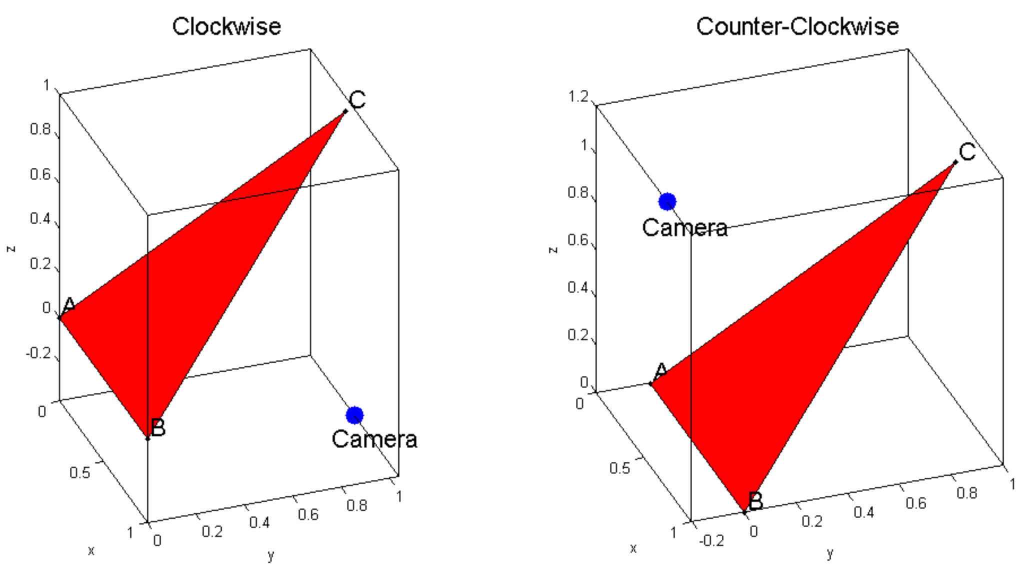
2. Given the vectors **u** and **v** below, find two **unit** vectors that are orthogonal to both.

   (a)* $\mathbf{u} = \langle 1, -1, 1\rangle$ and $\mathbf{v} = \langle -2, 3, 0\rangle$.

   (b) $\mathbf{u} = \langle 2, -1, 1\rangle$ and $\mathbf{v} = \langle -2, 3, 6\rangle$.

3. (**MATLAB®**): Given $\mathbf{u} = \langle 3, -1, 2\rangle$ and $\mathbf{v} = \langle 2, 2, 3\rangle$. Use MATLAB® to plot both vectors in 3D. On the same graph plot $\mathbf{u}\times\mathbf{v}$ and $\mathbf{v}\times\mathbf{u}$. Choose a viewing angle with `view(Az,El)` and set `axis equal;` so both cross products look orthogonal to both vectors. You can start with `PlottingCrossProducts.m` from the Chapter 2 program repository.

4. (**MATLAB®**): Grab the MATLAB® file `Culling_Starter.m` from the Chapter 2 program repository. Run it. It will give you the points A, B, and C which form a triangle. It also plots a camera point. Your task is to edit the program (the last few lines) so that the title of the graph tells whether the points appear in a clockwise or counter-clockwise fashion from the perspective of the camera.



2.5 Lines in 2D and 3D

You are probably accustomed to expressing a line in 2D as $y = mx + b$. While that works fine in 2D it will not work in 3D. Here we study **parametric equations** for describing lines in two and three dimensions. We used parametric equations when we plotted circles and ellipses in past exercises. Here we will use them to define lines. We start with lines in 3D because that is where the parametric representation is essential. However, there are good reasons to use parametric equations for lines in 2D as well. You'll see that later in this section.

Parametric Equations for a Line in 3D

When seeking the equation for a line in 3D we need a point and a direction vector.

A line L parallel to the vector $\mathbf{v} = \langle a,\ b,\ c\rangle$ and passing through the point $P(x_1,\ y_1,\ z_1)$ is represented by the **parametric equations**

$$x = x_1 + at, \quad y = y_1 + bt, \quad z = z_1 + ct \quad \text{for} \quad t \in (-\infty, \infty). \tag{2.21}$$

The vector $\mathbf{v}$ is called the **direction vector** and a, b, c are called the **direction numbers**.

♣ *Mini-Proof:* If $Q(x,\ y,\ z)$ is a point on L then $\overrightarrow{PQ} = \langle x - x_1,\ y - y_1,\ z - z_1\rangle$ is parallel to $\mathbf{v} = \langle a,\ b,\ c\rangle$. This means that $\overrightarrow{PQ} = t\,\mathbf{v}$ where t is a scalar. Therefore,

$$\begin{array}{lcl} x - x_1 = at & \iff & x = x_1 + at \\ y - y_1 = bt & \iff & y = y_1 + bt \\ z - z_1 = ct & \iff & z = z_1 + ct \end{array}$$

and the second column gives us the equations in (2.21) for all points on the line. ♣.

- **Example, Finding a line from a point and a direction vector:** Determine the parametric equations for the line through the point (3, -2, 0) parallel to the vector $\mathbf{v} = \langle -2,\ 0,\ -1\rangle$.

  Answer: $x = 3 - 2t, \quad y = -2, \quad z = -t$

- **Example, Finding a line from two points:** Determine the parametric equations for the line through $P(-2,\ 1,\ 0)$ and $Q(1,\ 3,\ 5)$.

  Answer: You have two points on the line (you only need one) but you also need a direction vector $\mathbf{v}$. In this case $\mathbf{v} = \overrightarrow{PQ} = \langle 1,\ 3,\ 5\rangle - \langle -2,\ 1,\ 0\rangle = \langle 3,\ 2,\ 5\rangle$. Choosing the point (-2, 1, 0) as the point for describing the line, the parametric equations are

  $$x = -2 + 3t, \quad y = 1 + 2t, \quad z = 5t.$$

  Note: You can use either of the two points and get an equivalent set of equations for the line. You could also use $\overrightarrow{QP}$ and the direction numbers would switch sign resulting in still another equivalent set of equations for the same line.

Parallel Lines: Two lines are parallel if their direction vectors are parallel.

- **Examples, Parallel Lines:** Find the direction vectors of the two lines and determine whether the lines are parallel or not.

  1. $L_1: \quad x = 3 - 4t, \quad y = 3, \quad z = 1 - t$
     $L_2: \quad x = 2 + 2s, \quad y = 3 + 2s, \quad z = 1 - s$
     Answer: The direction vector for L_1 is $\langle -4,\ 0,\ -1\rangle$. The direction vector for L_2 is $\langle 2,\ 2,\ -1\rangle$. Since the direction vectors are not parallel, the lines are not parallel.

  2. $L_1: \quad x = 3, \quad y = 3 + t, \quad z = 4 - 2t$
     $L_2: \quad x = 7, \quad y = 8 + 3s, \quad z = 1 - 6s$
     Answer: The direction vector for L_1 is $\langle 0,\ 1,\ -2\rangle$. The direction vector for L_2 is $\langle 0,\ 3,\ -6\rangle$. Since the direction vectors are parallel, the lines are parallel.

Intersection of Lines in 3D: Unlike 2D, it is quite possible that two lines in 3D do not intersect and are not parallel. They could just lie in different planes. To find the intersection of two lines in space, you must solve a system of 3 equations in 2 unknowns. As such, you have to get pretty lucky to get a valid solution.

- **Example, two lines in 3D that DO NOT intersect:**

  Determine if the lines intersect, if so, find the point of intersection and the angle of intersection.

  $L_1: \quad x = 2 + t, \quad y = 3, \quad z = 1 + 4t$

  $L_2: \quad x = 5 - s, \quad y = 3 + 2s, \quad z = 9 + 2s$

  Answer You need to determine if there are values for t and s where (x, y, z) are the same for both lines. So we need to solve these three equations for the two unknown values of s and t.

$$\begin{array}{lrcl} \text{for } x\text{:} & 2+t & = & 5-s \\ \text{for } y\text{:} & 3 & = & 3+2s \\ \text{for } z\text{:} & 1+4t & = & 9+2s \end{array} \sim \begin{array}{rcl} t+s & = & 3 \\ -2s & = & 0 \\ 4t-2s & = & 8 \end{array} \sim \begin{bmatrix} 1 & 1 \\ 0 & -2 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 8 \end{bmatrix}$$

  Expressing these equations in augmented matrix form, you can use Gaussian Elimination to solve for the variables (or have software do it for you but be careful of the *no solution* problems that can occur).

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & -2 & 0 \\ 4 & -2 & 8 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & -2 & 0 \\ 0 & -6 & -4 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & 1 & 0 \\ 0 & -6 & -4 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & -4 \end{array}\right]$$

  The last equation says that $0t + 0s = -4$. There are no solutions to such an equation so the lines **do not intersect**. Notice, however, the lines are not parallel.

  Warning: If you try to solve this system in MATLAB® , it will not give you a warning and will produce a *least squares solution*. This is not a real solution. It is the best of all non-solutions. There will be no warning, so you should check that the solution you get is actually a solution by plugging it into the original system of equations.

- **Example, two lines in 3D that DO intersect:**

Determine if the lines intersect, if so, find the point of intersection and the angle of intersection.

$L_1: \quad x = 2 + t, \quad y = 3, \quad z = 1 + 4t$

$L_2: \quad x = 4 - s, \quad y = 3 + 2s, \quad z = 9 + 2s$

Answer: You need to determine if there are values for t and s where (x, y, z) are the same for both lines. So we need to solve these three equations for the two unknown values of s and t.

$$\begin{array}{lrcl} \text{for } x: & 2+t & = & 4-s \\ \text{for } y: & 3 & = & 3+2s \\ \text{for } z: & 1+4t & = & 9+2s \end{array} \sim \begin{array}{rcl} t+s & = & 2 \\ -2s & = & 0 \\ 4t-2s & = & 8 \end{array} \sim \begin{bmatrix} 1 & 1 \\ 0 & -2 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 8 \end{bmatrix}$$

Expressing these equations in augmented matrix form, you can use Gaussian Elimination to solve for the variables (or have software do it for you but be careful of the *no solution* problems that can occur).

$$\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & -2 & 0 \\ 4 & -2 & 8 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & -2 & 0 \\ 0 & -6 & 0 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & -6 & 0 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

The second equation says that $s = 0$, plugging this into the first equation gives $t = 2$. Notice when $t = 2$ and $s = 0$ in the parametric equations of the lines, they both result in $x = 4$, $y = 3$, and $z = 9$. **The point of intersection is (4,3,9).**

Note: If you try to solve this system in MATLAB® it will produce a valid solution. The solution can be verified by plugging it into the original system of equations.

To find the angle of intersection, you must find the angle between the two direction vectors $\mathbf{u} = \langle 1, 0, 4 \rangle$ and $\mathbf{v} = \langle -1, 2, 2 \rangle$. Using equation (2.15) for the angle between two vectors, you get

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \frac{7}{\sqrt{17}\,\sqrt{9}} = \frac{7}{3\sqrt{17}}.$$

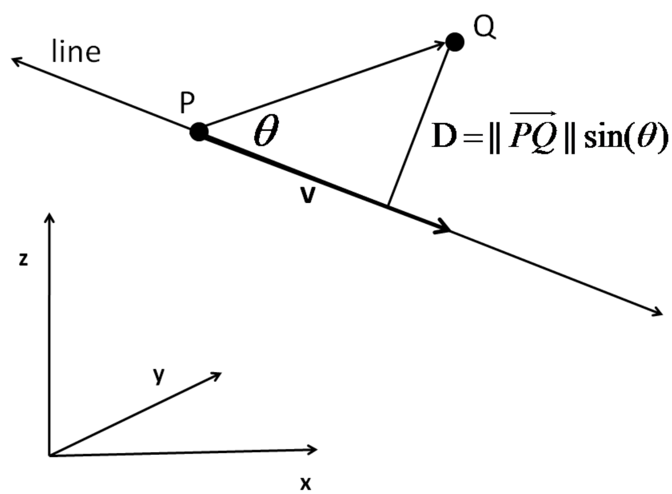
Therefore $\theta = \arccos\left(\frac{7}{3\sqrt{17}}\right) \approx 0.97$ radians or 55.5^o.

<u>**Distance Between a point and a line:**</u> Given a line with direction vector $\mathbf{v}$ and a point Q not on the line, then the distance between Q and the line is given by

$$D = \frac{||\overrightarrow{PQ} \times \mathbf{v}||}{||\mathbf{v}||} \tag{2.22}$$

where P is a any point on the line.

♣ *Mini-Proof:* As seen in the figure below, the distance from Q to the line is given by $D = ||\overrightarrow{PQ}||\sin(\theta)$. Since $||\overrightarrow{PQ} \times \mathbf{v}|| = ||\overrightarrow{PQ}||\,||\mathbf{v}||\,\sin(\theta)$, then $\dfrac{||\overrightarrow{PQ} \times \mathbf{v}||}{||\mathbf{v}||} = ||\overrightarrow{PQ}||\,\sin(\theta) = D$. ♣



- **Example:** Find the distance from the point $Q(-1, 0, 3)$ to the line defined by

$$x = 3 - t, \quad y = -2 + t, \quad z = 5 - 3t$$

Answer: The direction vector of the line is given by $\mathbf{v} = \langle -1, 1, -3\rangle$ and a point on the line is $P(3, -2, 5)$. Therefore, $\overrightarrow{PQ} = \langle -4, 2, -2\rangle$, and

$$\overrightarrow{PQ} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -4 & 2 & -2 \\ -1 & 1 & -3 \end{vmatrix} = -4\mathbf{i} - 10\mathbf{j} - 2\mathbf{k} = \langle -4, -10, -2\rangle\,,$$

then using equation (2.22),

$$D = \frac{||\overrightarrow{PQ} \times \mathbf{v}||}{||\mathbf{v}||} = \frac{||\,\langle -4, -10, -2\rangle\,||}{||\,\langle -1, 1, -3\rangle\,||} = \frac{\sqrt{\mathbf{120}}}{\sqrt{\mathbf{11}}} \approx \mathbf{3.303}$$

MATLAB® Demonstration - Plotting Lines in 3D

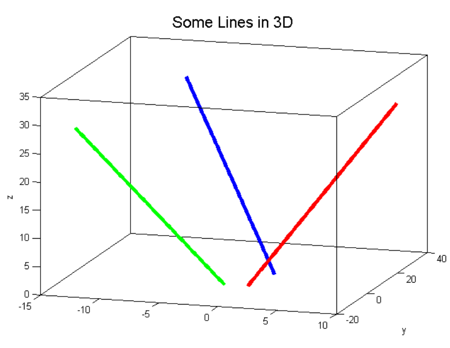
The `plot3` command is used to create a function file called `Plotline3d.m`. We use this to plot some lines given a point and a direction vector with the file `Plotting_Lines.m`. Both files are available in the Chapter 2 program repository.

MATLAB® Function File: `Plotline3d.m`

```
function  [] = Plotline3d(v,P,tbounds,whatcolor);
% This is the function Plot3d(v,P,tbounds,whatcolor)
% plots a line through P, parallel to v.
% tbounds = [tstart, tfinish]
% whatcolor: 'black', 'red', 'green', 'cyan', 'blue' etc.
% Uses MATLABS 'plot3'
  t = linspace(tbounds(1),tbounds(2),100);
      % = vector of 100 points between t1 and t2
  x = P(1) + v(1).*t;
  y = P(2) + v(2).*t;
  z = P(3) + v(3).*t;
  plot3(x,y,z,'color',whatcolor,'linewidth',4);
```

MATLAB® code: `Plotting_Lines.m`

```
%% Plotting_Lines.m
% Here we plot a line in 3D using the
% Plotline3d function we wrote
clc; clf; clear;

%% Let's plot some lines.
v1 = [1 2 3]; P1 = [0 0 0];
v2 = [-1 2 3]; P2 = [2 2 2];
v3 = [-1 -2 3]; P3 = [-2 0 0];
tbounds = [0, 10];
Plotline3d(v1,P1,tbounds,'red'); hold on;
Plotline3d(v2,P2,tbounds,'blue');
Plotline3d(v3,P3,tbounds,'green');
title('Some Lines in 3D','fontsize',16);
xlabel('x'); ylabel('y'); zlabel('z');
box on; view(17,18);
```



Parametric Equations for a Line in 2D

Defining the parametric equations for a line in 2D is exactly the same as in 3D with one less dimension.

A line L parallel to the vector $\mathbf{v} = \langle a, b\rangle$ and passing through the point $P(x_1, y_1)$ is represented by the **parametric equations**

$$x = x_1 + at, \quad y = y_1 + bt \quad \text{for} \quad t \in (-\infty, \infty). \tag{2.23}$$

The vector $\mathbf{v}$ is called the **direction vector**. As in our 3D version of this representation, two lines with the parallel direction vectors are parallel.

Intersection of Lines in 2D: We can find the intersection and angle of intersection of 2 lines in 2D as we did with lines in 3D except now there are 2 equations and 2 unknowns. This means there is a much better chance there will be a unique solution which defines the point of intersection. The other nice thing about 2D is if the lines are not parallel, then they will intersect (possibly at infinitely many points).

- **Example, two lines in 2D that DO NOT intersect at a unique point:**
  Determine if the lines intersect, if so, find the point of intersection and the angle of intersection.

  $$L_1: \quad x = t+1, \quad y = -3t+2$$
  $$L_2: \quad x = 3s+3, \quad y = -9s-4$$

  Answer: If you notice the direction vector for L_1 is $\langle 1, -3\rangle$ and the direction vector for L_2 is $\langle 3, -9\rangle$. These direction vectors are parallel but it is difficult to tell whether or not they are the same line. Suppose you didn't notice the parallel direction vectors. You would need to solve the equations

  $$\begin{array}{lrcl} \text{for } x: & t+1 & = & 3s+3 \\ \text{for } y: & -3t+2 & = & -9s-4 \end{array} \sim \begin{array}{rcr} t-3s & = & 2 \\ -3t+9s & = & -6 \end{array} \sim \begin{bmatrix} 1 & -3 \\ -3 & 9 \end{bmatrix}\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 2 \\ -6 \end{bmatrix}$$

  The determinant is 9-9 $= 0$ so there is not a unique solution for s and t. Performing Gaussian Elimination on this system yields

  $$\left[\begin{array}{rr|r} 1 & -3 & 2 \\ -3 & 9 & -6 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & -3 & 2 \\ 0 & 0 & 0 \end{array}\right].$$

  The last equation tells us there are infinitely many solutions so we know these two lines are the same. Since this means there is an intersection, we can get the angle of intersection by finding the angle between the two direction vectors by

  $$\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \frac{30}{\sqrt{10}\,\sqrt{90}} = \frac{30}{\sqrt{900}} = 1.$$

  Since the cosine of the angle is 1 it means that the angle between them is zero. We should have expected this because we know they are parallel. In fact, they are the same line.

- **Example, two lines in 2D that DO intersect:**
  Determine if the lines intersect, if so, find the point of intersection and the angle of intersection.

  $$L_1: \quad x = t+1, \quad y = 3t+2$$
  $$L_2: \quad x = s+4, \quad y = -4s+4$$

  Answer: You need to determine if there are values for t and s where (x, y) are the same for both lines. So we need to solve these two equations for the two unknown values of s and t.

  $$\begin{array}{lrcl} \text{for } x: & t+1 &=& s+4 \\ \text{for } y: & 3t+2 &=& -4s+4 \end{array} \sim \begin{array}{rcl} t-s &=& 3 \\ 3t+4s &=& 2 \end{array} \sim \begin{bmatrix} 1 & -1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

  The determinant is 4+3 = 7 so there is a unique solution for s and t. You can use Gaussian Elimination or the inverse of the matrix to get the solutions $t = 2$ and $s = -1$. So the point of intersection occurs when $t = 2$ in L_1 or $s = -1$ in L_2

  $L_1: \; x = 2+1, \; y = 3(2)+2 \quad \to \quad (x, y) = (3, 8)$
  $L_2: \; x = -1+4, \; y = -4(-1)+4 \quad \to \quad (x, y) = (3, 8)$

  The angle of intersection is the angle between the two direction vectors $\mathbf{u} = \langle 1, 3\rangle$ and $\mathbf{v} = \langle 1, -4\rangle$. You get $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \frac{-11}{\sqrt{10}\,\sqrt{17}}$. Therefore $\theta = \arccos\left(\frac{-11}{\sqrt{10}\,\sqrt{17}}\right) \approx 2.57$ radians or 147.5^o.

Directed Line Segments

The parametric equations for a **directed line segment** that starts at P and ends at Q are determined by the set of points defined by

$$P + t\,\overrightarrow{PQ} \qquad \text{or} \qquad P + t\,(Q-P) \quad \text{for} \quad t \in [0,1] \tag{2.24}$$

Notice, when $t = 0$, the line starts at P, and when $t = 1$, the line ends at Q. Equation (2.24) is valid in 2D and 3D. The specific parametric equations in 3D are

$$x = p_1 + t(q_1 - p_1), \quad y = p_2 + t(q_2 - p_2), \quad z = p_3 + t(q_3 - p_3), \qquad \text{for} \quad t \in [0,1]. \tag{2.25}$$

where $P = (p_1, p_2, p_3)$ and $Q = (q_1, q_2, q_3)$. The specific parametric equations in 2D are

$$x = p_1 + t(q_1 - p_1) \quad \text{and} \quad y = p_2 + t(q_2 - p_2) \quad \text{for} \quad t \in [0,1]. \tag{2.26}$$

where $P = (p_1, p_2)$ and $Q = (q_1, q_2)$.

- **Example:**
  Find the parametric equations for the line segment that starts at P(2,5) and ends at Q(7,-2).

  Answer: $$\begin{array}{l} x = 2 + t(7-2) \\ y = 5 + t(-2-5) \end{array} \quad \to \quad \begin{array}{l} x = 2 + 5t \\ y = 5 + -7t \end{array} \qquad \text{for } t \in [0,1]$$

  Notice, when $t = 0$, (x, y) = (2,5) = P, and when $t = 1$, (x, y) = (7,-2) = Q as it should.

Chapter 2.5 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. **Parametric Equations for Lines in 3D:**
   Find the parametric equations for the line with the given properties.

   (a)* The line passes through the point (2,3,4) and is parallel to the vector $\mathbf{v} = \langle 1, -2, 3 \rangle$.

   (b) The line passes through the points (2,3,4) and (-1, 0, 4).

   (c)* The line passes through the point (2,3,4) and is parallel to the xz-plane and the yz-plane.

   (d) The line passes through the point (2,3,4) and is perpendicular to $\mathbf{u} = \langle 1, -1, 1 \rangle$ **and** $\mathbf{v} = \langle -2, 3, 0 \rangle$.
   Hint: Find $\mathbf{u} \times \mathbf{v}$ to get a vector that is perpendicular to both $\mathbf{u}$ and $\mathbf{v}$.

2. **Intersection of Lines in 3D:** Determine if the two lines (L_1 and L_2) intersect.
   If they do, where do they intersect and what is the angle of intersection?

   (a)* $L_1: \ x = 4t + 2, \ y = 3, \ z = -t + 1$
   $L_2: \ x = 2s + 2, \ y = 2s + 3, \ z = s + 1$

   (b) $L_1: \ x = 2t, \ y = t + 1, \ z = 3t - 2$
   $L_2: \ x = 3s - 5, \ y = s, \ z = -2s + 10$

   (c)* $L_1: \ x = 3t, \ y = -t + 2, \ z = t - 1$
   $L_2: \ x = 4s + 1, \ y = s - 2, \ z = -3s - 3$

   (d) $L_1: \ x = t + 2, \ y = 3t - 3, \ z = -t + 1$
   $L_2: \ x = -2s, \ y = -5s + 10, \ z = -5s - 3$

3. **Distance between a point and a line:**

   (a)* Find the distance between the point $Q(1, 5, -2)$ and the line $L: x = 4t - 2, \ y = 3, \ z = -t + 1$.

   (b) Find the distance between the point $Q(1, 5, -2)$ and the line with direction vector $\mathbf{v} = \langle 1, 2, -3 \rangle$ through the point $(0, 2, 4)$.

4. **Intersection of Lines in 2D:** Determine if the two lines (L_1 and L_2) intersect.
   If they do, where do they intersect and what is the angle of intersection?

   (a)* $L_1: \ x = 4t + 2, \ y = 6t + 3$
   $L_2: \ x = s + 1, \ y = 2s + 4$

   (b) $L_1: \ x = t - 2, \ y = -t + 5$
   $L_2: \ x = 2s + 1, \ y = s - 4$

   (c)* $L_1: \ x = t - 2, \ y = -t + 5$
   $L_2: \ x = -3s + 1, \ y = 3s - 4$

   (d) $L_1: \ x = t - 2, \ y = -t + 5$
   $L_2: \ x = -3s - 2, \ y = 3s + 5$

5. **Parametric Equations for Line Segments:** Find the parametric equations for the line that starts at P and terminates at Q.

   (a)* $P(3, 2, 1)$ and $Q(-2, -1, 1)$

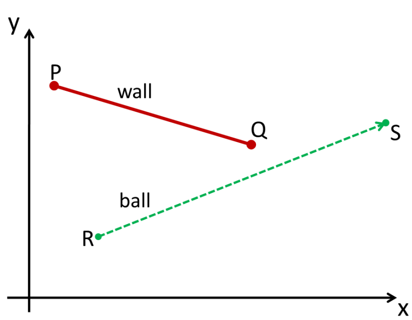
   (b) $P(0, 2, 5)$ and $Q(7, 2, 6)$

   (c)* $P(1, 2)$ and $Q(-2, 5)$

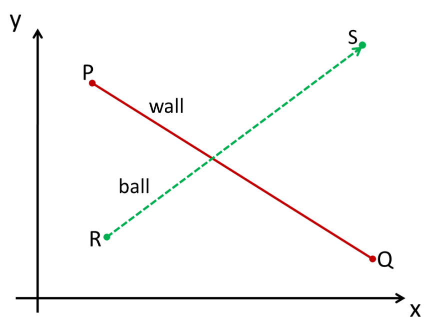
   (d) $P(-1, -2)$ and $Q(2, 7)$

6. **Intersection of 2 Line Segments - Collision Detection:**
   Here we check to see if two lines intersect between two given points. Suppose there is a wall from point P to point Q and you have a *ball* going from point R to point S. You want to know if the ball hits the wall. Equivalently, you want to know if the directed line segment $\overrightarrow{RS}$ intersects the directed line segment $\overrightarrow{PQ}$. It is not sufficient to know whether the lines defined by P & Q and R & S intersect. We want to know if the lines intersect between P & Q and R & S. And, if they intersect, where does the intersection occur?

No Hit



Hit



There are a few ways to do this and here is one way. The directed line segments $\overrightarrow{PQ}$ and $\overrightarrow{RS}$ can be defined as follows.

$$\begin{aligned}\overrightarrow{PQ} &= P + t\,\overrightarrow{PQ} = P + t(Q - P) \quad \text{for} \quad t \in [0, 1] \\ \overrightarrow{RS} &= R + s\,\overrightarrow{RS} = R + s(S - R) \quad \text{for} \quad s \in [0, 1]\end{aligned}$$

If you can solve these equations for t and s and both are between 0 and 1, then the line segments intersect. If either one (t or s) is not between 0 and 1, then they do not intersect.

   (a)* Determine if, and where, the ball going from $R(3, 1)$ to $S(5, 11)$ hits the wall defined by $P(1, 8)$ and $Q(10, 2)$.

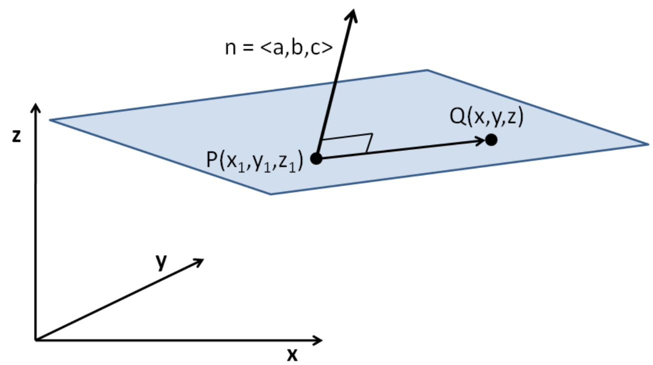
   (b) Determine if, and where, the ball going from $R(-2, 8)$ to $S(6, 4)$ hits the wall defined by $P(2, 2)$ and $Q(14, 4)$.

2.6 Planes

A plane is a flat two-dimensional surface. When defining a plane in 3-space, you need a point through which the plane passes and a vector which is perpendicular to the plane. A vector that is perpendicular to a plane (or surface) it is called a **normal** vector. A plane is defined in terms of a point and a normal vector.

<u>**Equations for a Plane**</u>

If a plane goes through the point $P(x_1,\ y_1,\ z_1)$ and has a normal vector $\mathbf{n} = \langle a,\ b,\ c\rangle$, then for any other point $Q(x,\ y,\ z)$ on the plane, it must be true that $\mathbf{n} \cdot \overrightarrow{PQ} = 0$. This requirement leads to the equations that describe a plane.



$$
\begin{aligned}
\mathbf{n} \cdot \overrightarrow{PQ} &= 0 && \\
a(x - x_1) + b(y - y_1) + c(z - z_1) &= 0 && \text{standard equation} \\
ax + by + cz + d &= 0 && \text{general equation} \\
z &= (-d - ax - by)/c && \text{function form } z = f(x, y) \\
y &= (-d - ax - cz)/b && \text{function form } y = g(x, z) \\
x &= (-d - by - cz)/a && \text{function form } x = h(y, z)
\end{aligned} \tag{2.27}
$$

In these equations, $d = -ax_1 - by_1 - cz_1$.

- **Example, Finding a plane from a point and a normal vector:** Find the general equation for a plane that goes through the point (1, -2, 3) with normal vector $\mathbf{n} = \langle 4,\ 5,\ -6\rangle$.

  Answer: Start with the standard form and then convert it to general by

$$
\begin{aligned}
4(x - 1) + 5(y + 2) - 6(z - 3) &= 0 && \text{standard equation} \\
4x - 4 + 5y + 10 - 6z + 18 &= 0 && \text{intermediate step} \\
4x + 5y - 6z + 24 &= 0 && \text{general equation}
\end{aligned}
$$

- **Example, Finding a plane from 3 points:** Find the standard form of the plane through $P(2,\ 1,\ 1)$, $Q(0,\ 4,\ 1)$, $R(-2,\ 1,\ 4)$.

  Answer: We need a point on the plane (we have three) and a normal vector. To get the normal vector $\mathbf{n}$ we set $\mathbf{n} = \overrightarrow{PQ} \times \overrightarrow{PR}$ which will be orthogonal to both vectors that form the plane. In our case, $\overrightarrow{PQ} = \langle -2,\ 3,\ 0\rangle$, $\overrightarrow{PR} = \langle -4,\ 0,\ 3\rangle$, and

$$\overrightarrow{PQ} \times \overrightarrow{PR} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -2 & 3 & 0 \\ -4 & 0 & 3 \end{vmatrix} = 9\,\mathbf{i} + 6\,\mathbf{j} + 12\,\mathbf{k} = \langle 9,\ 6,\ 12\rangle = \mathbf{n}.$$

  Using the point $P(2,\ 1,\ 1)$ as the point,

$$\begin{array}{rcll} 9(x-2)+6(y-1)+12(z-1) & = & 0 & \text{standard equation} \\ 9x-18+6y-6+12z-12 & = & 0 & \text{intermediate step} \\ 9x+6y+12z-36 & = & 0 & \text{general equation} \end{array}$$

  Using the point $Q(0,\ 4,\ 1)$ as the point,

$$\begin{array}{rcll} 9(x-0)+6(y-4)+12(z-1) & = & 0 & \text{standard equation} \\ 9x+6y-24+12z-12 & = & 0 & \text{intermediate step} \\ 9x+6y+12z-36 & = & 0 & \text{general equation} \end{array}$$

  Using the point $R(-2,\ 1,\ 4)$ as the point,

$$\begin{array}{rcll} 9(x+2)+6(y-1)+12(z-4) & = & 0 & \text{standard equation} \\ 9x+18+6y-6+12z-48 & = & 0 & \text{intermediate step} \\ 9x+6y+12z-36 & = & 0 & \text{general equation} \end{array}$$

  Notice: It doesn't matter which point you choose, you get the same general equation.

- **Another Example:** Find the plane through the point $P(2,\ 1,\ -4)$ and perpendicular to the line described by the parametric equations

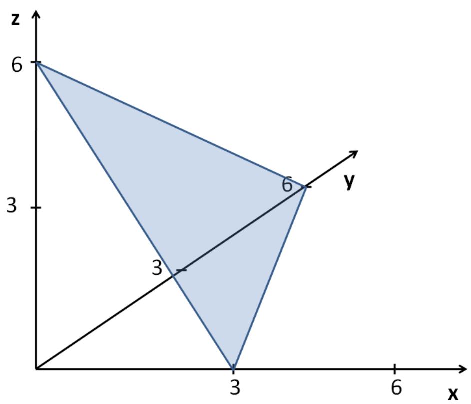
$$L: \quad x = 2 - t, \quad y = 3, \quad z = 3t.$$

  Answer: We have a point, so all we need is a normal vector. The direction vector for the given line is $\langle -1,\ 0,\ 3\rangle$. Letting this be the normal vector to the plane we get

$$\begin{array}{rcll} -1(x-2)+0(y-1)+3(z+4) & = & 0 & \text{standard equation} \\ -x+2+3z+12 & = & 0 & \text{intermediate step} \\ -x+3z+14 & = & 0 & \text{general equation} \end{array}$$

  Notice: The line is parallel to the xz-plane because y is constant. Therefore the plane is parallel to the y-axis. (There is no y-intercept). Try to picture this in your head.

Sketching a Plane: You can get a decent sketch of a plane by plotting the x, y, and z-intercepts then connecting the dots.



- **Example:**

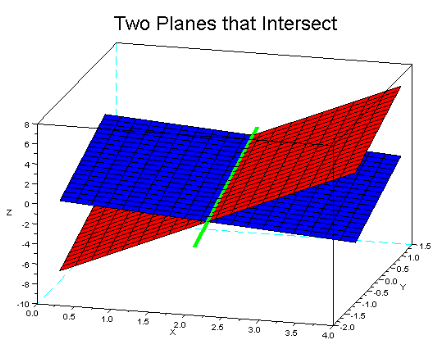
Sketch $4x + 2y + 2z - 12 = 0$ using intercepts.

- The x intercept is where both y, and z are zero. This results in the equation $4x - 12 = 0$ or $x = 3$.

- The y intercept is where both x, and z are zero. This results in the equation $2y - 12 = 0$ or $y = 6$.

- The z intercept is where both x, and y are zero. This results in the equation $2z - 12 = 0$ or $z = 6$.

The Angle Between Two Planes: If two planes have normal vectors given by $\mathbf{n_1}$ and $\mathbf{n_2}$, then the angle of intersection $(0 \leq \theta \leq \pi/2)^*$ is given by



$$\cos(\theta) = \frac{|\mathbf{n_1} \cdot \mathbf{n_2}|}{||\mathbf{n_1}||\ ||\mathbf{n_2}||} \tag{2.28}$$

$$\theta = \arccos\left(\frac{|\mathbf{n_1} \cdot \mathbf{n_2}|}{||\mathbf{n_1}||\ ||\mathbf{n_2}||}\right) \tag{2.29}$$

- The planes are perpendicular if $\mathbf{n_1} \cdot \mathbf{n_2} = 0$.
- The planes are parallel if $\mathbf{n_2} = c\,\mathbf{n_1}$.

- **Example:** Given the two planes defined below, find the angle of intersection.

$$\text{Plane 1:}\quad 3x - y + 2z - 12 = 0 \qquad \text{Plane 2:}\quad x - 2z + 3 = 0$$

Answer: The two normal vectors are given by $\mathbf{n}_1 = \langle 3,\ -1,\ 2\rangle$ and $\mathbf{n}_2 = \langle 1,\ 0,\ -2\rangle$. The angle of intersection is given by

$$\theta = \arccos\left(\frac{|\mathbf{n_1} \cdot \mathbf{n_2}|}{||\mathbf{n_1}||\ ||\mathbf{n_2}||}\right) = \arccos\left(\frac{1}{\sqrt{14}\,\sqrt{5}}\right) = \arccos\left(\frac{1}{\sqrt{70}}\right) \approx 1.45 \text{ radians or } 83.1^o$$

---

*In these equations, the absolute value of the dot product ($|\mathbf{n_1} \cdot \mathbf{n_2}|$) is used to ensure that we get the smaller of the two angles of intersection.

<u>**Getting the Line of the Intersection of Two Planes:**</u> If two planes intersect, they intersect in a line.

Method 1: (hand calculations) Find the general solution to a system of linear equations.

Method 2: (software calculations) Find one particular solution and get a direction vector by $\mathbf{v} = \mathbf{n}_1 \times \mathbf{n}_2$.

- **Example:** Given the two planes below, find the parametric equations for the line of intersection.

$$\text{Plane 1:} \quad x + 3z - 10 = 0 \qquad \text{Plane 2:} \quad 2x + y + 5z + 3 = 0$$

Answer with Method 1: We must find all points on the intersection of both planes. This means we must find values of x, y and z that satisfy both plane equations or

$$\begin{array}{rcl} x + 3z - 10 & = & 0 \\ 2x + y + 5z + 3 & = & 0 \end{array} \quad \sim \quad \begin{array}{rcl} x + 3z & = & 10 \\ 2x + y + 5z & = & -3 \end{array} \quad \sim \quad \begin{bmatrix} 1 & 0 & 3 \\ 2 & 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 10 \\ -3 \end{bmatrix}$$

Expressing these equations in augmented matrix form, you can use Gaussian Elimination to solve for the variables. You will get infinitely many solutions provided the planes actually intersect.

$$\left[\begin{array}{ccc|c} 1 & 0 & 3 & 10 \\ 2 & 1 & 5 & -3 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 3 & 10 \\ 0 & 1 & -1 & -23 \end{array}\right]$$

The second equation states that $y - z = -23$. If we let z be the free variable, then $y = -23 + z$ and the first equation becomes $x = 10 - 3z$. Letting $z = t$ be the free variable, the general solution is given by

$$x = 10 - 3t, \qquad y = -23 + t, \qquad z = t, \quad \text{for} \quad t \in (-\infty, \infty)$$

This general solution represents the parametric equations for the line.

Answer with Method 2: Set up the system of equations as in Method 1 and get one particular solution. If a system has infinitely many solutions, software can usually get one of them for you. MATLAB® obtains $\left[0, \frac{-59}{3}, \frac{10}{3}\right]^T$. The normal vectors for the planes are given by $\mathbf{n}_1 = \langle 1, 0, 3 \rangle$ and $\mathbf{n}_2 = \langle 2, 1, 5 \rangle$. The direction vector for the line of intersection $\mathbf{v}$ is

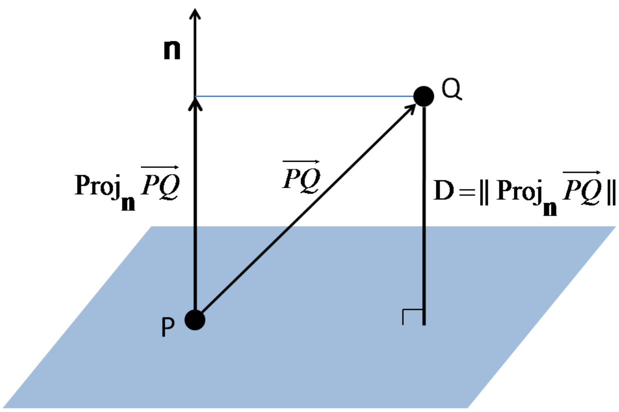
$$\mathbf{v} = \mathbf{n_1} \times \mathbf{n_2} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & 3 \\ 2 & 1 & 5 \end{vmatrix} = -3\mathbf{i} + \mathbf{j} + \mathbf{k} = \langle -3, 1, 1 \rangle$$

Now we have a direction vector $\mathbf{v} = \langle -3, 1, 1 \rangle$ and a point $\left(0, \frac{-59}{3}, \frac{10}{3}\right)^T$. So the parametric equations for the line of intersection are

$$x = -3s, \qquad y = \frac{-59}{3} + s, \qquad z = \frac{10}{3} + s, \quad \text{for} \quad s \in (-\infty, \infty)$$

The two methods seldom produce the same parametric equations. However, they will always trace out the same line and have parallel direction vectors.

Distance between a point and a plane



The distance between a plane through P with $\mathbf{n} = \langle a, b, c\rangle$ and a point Q is

$$D = ||\text{proj}_{\mathbf{n}}\overrightarrow{PQ}|| = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||} \tag{2.30}$$

where P is a point in the plane.

♣ *Mini-Proof:* As seen in the figure above, $D = ||\text{Proj}_{\mathbf{n}}\overrightarrow{PQ}|| = \|\frac{\overrightarrow{PQ}\cdot\mathbf{n}}{||\mathbf{n}||^2}\ \mathbf{n}\| = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||^2}\ ||\mathbf{n}|| = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||}$.

Note on Collision Detection: If you want to see if Q has hit the plane, you want to check if $D = 0$. Unfortunately, the point Q might pass through the plane in a given time step without $D = 0$. However, if this happens, Proj$_{\mathbf{n}}\overrightarrow{PQ}$ will change sign. So, in addition to checking if $D = 0$ it also worth checking whether the sign of $\overrightarrow{PQ}\cdot\mathbf{n}$ changes during a given time step. If it has, then a collision has occurred.

- **Example, The distance between a point and a plane:** Find the distance between the point $Q(1, 2, 3)$ and the plane $4x - 2y + z - 6 = 0$.

  Answer: The normal to the plane is given by $\mathbf{n} = \langle 4, -2, 1\rangle$. To find a point on the plane, let $x = y = 0$, then $z = 6$ and $P = (0, 0, 6)$. Now, the vector $\overrightarrow{PQ} = \langle 1, 2, -3\rangle$. So

$$D = ||\text{proj}_{\mathbf{n}}\overrightarrow{PQ}|| = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||} = \frac{3}{\sqrt{21}}$$

- **Example, The distance between two parallel planes** Given the equations for two parallel planes, find the distance between them.

$$\text{Plane 1:}\quad 3x - y + 2z - 6 = 0 \qquad \text{Plane 2:}\quad 3x - y + 2z + 5 = 0$$

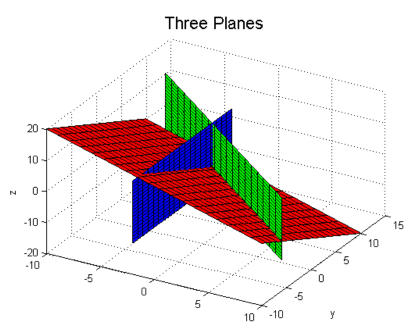
  Answer: Let Q be a point on Plane 2, $Q = (0, 5, 0)$. Now just find the distance from Q to Plane 1. The normal vector is given by $\mathbf{n} = \langle 3, -1, 2\rangle$ and a point on Plane 1 is $P(0, -6, 0)$. Now, the vector $\overrightarrow{PQ} = \langle 0, 11, 0\rangle$. So

$$D = ||\text{proj}_{\mathbf{n}}\overrightarrow{PQ}|| = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||} = \frac{11}{\sqrt{14}}$$

MATLAB® Demonstration - Plotting Planes: The `surf` command is used to create a function file called `PlotPlaneFunction.m`. We use this to plot some planes given a point and a normal vector with the file `Plotting_Planes.m`. Both files are available in the Chapter 2 program repository.

```
function [] = PlotPlaneFunction(n,P,xyzbounds,whatcolor)
  % n = normal vector; P = a point;
  % xybounds = [xmin, xmax, ymin, ymax, zmin, zmax]
  if norm(n) == 0 then           % can't have n = [0 0 0]
    error('n must be nonzero in PlotPlaneFunction')
    abort;
  end
  a = n(1); b = n(2); c = n(3);    % set a,b,c from normal
  x1 = P(1); y1 = P(2); z1 = P(3);    % set x1,y1,z1 from the point
  d = a*x1 + b*y1 + c*z1;   % find d  (= -d from the general form)
  xmin = xyzbounds(1); xmax = xyzbounds(2); % get x bounds
  ymin = xyzbounds(3); ymax = xyzbounds(4); % get y bounds.
  zmin = xyzbounds(5); zmax = xyzbounds(6); % get z bounds.
  xvec = linspace(xmin,xmax,20);  % nice trick
  yvec = linspace(ymin,ymax,20);  % to get 20 points
  zvec = linspace(zmin,zmax,20);  % from start to finish

  if c ≠ 0                  % if c is not 0
    [x y] = meshgrid(xvec,yvec);  % mesh out x and y
    z = (d - a*x - b*y)/c;        % define z on mesh
  elseif b ≠ 0               % if b doesn't = 0
    [x z] = meshgrid(xvec,zvec);  % mesh out x and z.
    y = (d - a*x - c*z)/b;        % define y on mesh.
  else                            % a must not be zero.
    [y z] = meshgrid(yvec,zvec);  % mesh out y and z.
    x = (d - b*y - c*z)/a;        % define on x mesh
  end
surf(x, y, z,'FaceColor', whatcolor); % plot surface.
```

```
%% This is Plotting_Planes.m
% It uses PlotPlaneFunction.m
clc; clf; clear;

%% Define Planes with Point and Normal
n1 = [2 2 2]; P1 = [0 0 0];
n2 = [2 2 0]; P2 = [1 3 -2];
n3 = [2 0 0]; P3 = [-2 -1 0];

%% Graph bounds [xmin,xmax,ymin,ymax,zmin,zmax]
xyzbounds =   [-10, 10, -10, 10, -10, 10];

%% Plot the Planes with PlotPlaneFunction
PlotPlaneFunction(n1,P1,xyzbounds,'red');
hold on; % don't erase the graph
PlotPlaneFunction(n2,P2,xyzbounds,'green');
PlotPlaneFunction(n3,P3,xyzbounds,'blue');
title('Three Planes','fontsize',16);
xlabel('x'); ylabel('y'); zlabel('z');
view(30,40);  % found by rotating the graph
```

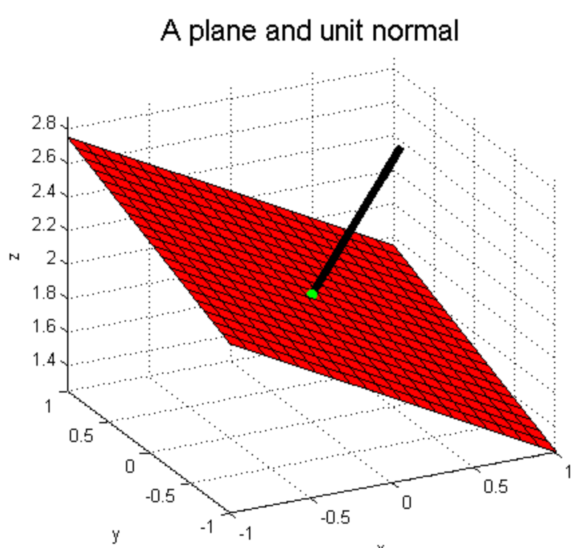


MATLAB® Demonstration - Plotting Planes and Normal Vectors

Here we combine the graph of a plane and it's unit normal emanating from a given point. This file (`Plot_Plane_And_Normal.m`) is available in the Chapter 2 program repository.

Example: Plot the plane through the point $P(0, 0, 2)$ with normal vector $\mathbf{n} = \langle 1, -0.5, 2 \rangle$. Additionally, plot the unit normal vector emanating from the point P. Rotate the figure to make sure the normal vector looks normal to the plane from different viewing perspectives.

```
% This is Plot_Plane_And_Normal.m
% It uses PlotPlaneFunction.m and vectoarrow.m
% It plots the plane and the unit normal emanating from a given point.
clc; clf; clear;  % clears console, figures, and variables.

n = [1 -.5 2]; % The normal vector to the plane.
P = [0 0 2];  % A point on the plane.
xyzbounds = [-1, 1, -1, 1, -1, 1]; % [xmin,xmax,ymin,ymax,zmin,zmax]
PlotPlaneFunction(n,P,xyzbounds,'red');  hold on;% plot the plane
plot3(P(1),P(2),P(3),'g.','markersize',20);   % plot the point P.

unitn = n/norm(n);  % find the unit normal
P2 = P + unitn;     % find the terminal point
vectarrow(P,P2,'black');   % plots unit normal with initial point P
xlabel('x'); ylabel('y'); zlabel('z');
axis equal;     % makes normal vectors look normal
view(-40,30);   % found by rotating the graph
title('A plane and unit normal','fontsize',16);
```



Chapter 2.6 Worksheet

Way back when solving systems of equations we came across the 3 x 3 system

$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

and we determined that there were no solutions. Allowing each equation to represent the graph of a plane, you can see that the planes are not parallel. Graph the three planes in such a way that you can ascertain why there is no point of intersection.

Chapter 2.6 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. Determine the equation for the plane with the given properties.

   (a)* The plane goes through the points $P(0,1,2)$, $Q(1,0,3)$, and $R(-2,3,4)$.

   (b) The plane goes through the point (0,0,6) and is perpendicular to the line given by $x = 1 - t$, $y = 2 + t$, $z = 4 - 2t$.

   (c)* The plane goes through the point (1,2,3) and is parallel to the xy-plane.

   (d) The plane passes through the points (2,2,1) and (-1,1,-1) and is perpendicular to the plane $2x - 3y + z = 3$.

2. Sketch the given plane using intercepts.

   (a)* $6x + 4y - 3z + 24 = 0$

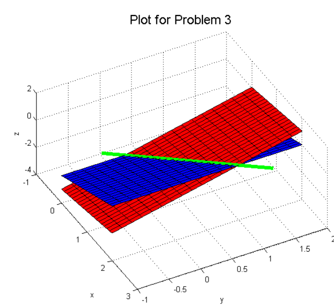
   (b) $x + y - 2z - 6 = 0$

3.* Consider the two planes $x + 2y - 2z = 4$ and $3x - 3z = 6$.

   (a) Find the angle of intersection.

   (b) Find the parametric equations for the line of intersection of the planes.

   (c) Use MATLAB® to plot both planes and the line you found in part (b). Does the line follow the intersection? Your graph should look something like the one here.

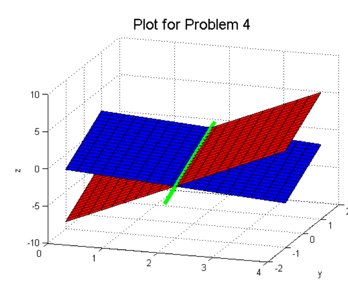


4. Consider the two planes $x - 4y + 2z = 0$ and $3x + 2y - z = 7$.

   (a) Find the angle of intersection.

   (b) Find the parametric equations for the line of intersection of the planes.

   (c) Use MATLAB® to plot both planes and the line you found in part (b). Does the line follow the intersection? Your graph should look something like the one here.



5. Calculate the distance between the given point and the given plane.

   (a)* The point is $Q(1,-1,1)$ and the plane is defined by $6x-3y+2z-8=0$.

   (b) The point is $Q(1,2,3)$ and the plane is defined by $3x+2y-5z+8=0$.

6. Planes and Spheres

   (a)* Below are the equations for a sphere and a plane. Use the formula for the distance between a point and a plane to determine if the sphere intersects the plane.

   $$\text{sphere: } (x-1)^2+(y+2)^2+z^2=4 \qquad \text{plane: } 2x-3y+z-2=0$$

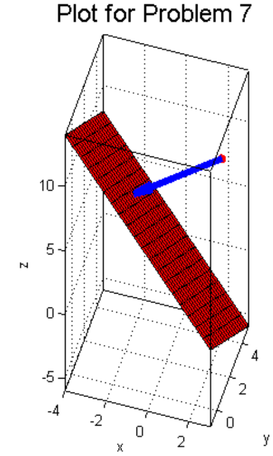
   (b) Find the standard equation of the sphere that is centered at $Q(8,-5,7)$ and is tangent to the plane $2x-y-2z+8=0$.

7.* (**MATLAB®**) Find the distance between the point (2,4,8) and the plane $2x+y+z=5$. Use MATLAB® to plot the point and the plane then draw the directed line segment between the point and the plane that has this distance as its length.

   Your plot should look like the one to the right.

   In order to make sure orthogonal objects appear orthogonal it is necessary to use the command `axis equal;` following the plot command.

Plot for Problem 7

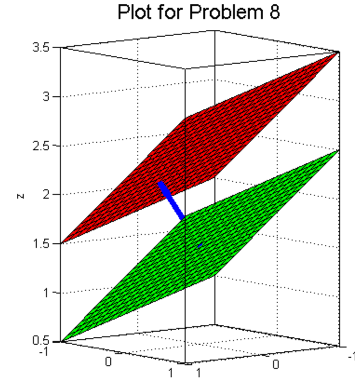


8.* (**MATLAB®**) Find the distance between the planes $x-3y+4z=10$ and $x-3y+4z=6$. Use MATLAB® to plot both planes and a directed line segment connecting the planes with length equal to this distance.

   Your graph should look like the one to the right.

   In order to make sure orthogonal objects appear orthogonal it is necessary to use the command `axis equal;` following the plot command.

Plot for Problem 8



2.7 Collision Detection and Response: Lines and Planes

In this section we see how to determine a bounce when an object moving in a straight line hits a plane. In order to determine this response, we must first determine where the straight line intersects the plane.

The Intersection of a Line and a Plane
To determine where a line intersects a plane, you substitute the parametric expressions for x, y, and z into the equation for the plane and solve for the parameter. Once that value is found you substitute the it into the equations for the line to get the point of intersection.

- **Example:** Determine the point of intersection of the line and plane given below.

$$\text{Line:} \quad x = 1 - t, \quad y = 4t, \quad z = 9 \qquad \text{Plane:} \quad 3x + 2y - 2z = 0$$

Answer: Put the expressions for x, y, and z, into the equation for the plane and solve for t.

$$\begin{aligned} 3x + 2y - 2z &= 0 \\ 3(1-t) + 2(4t) - 2(9) &= 0 \\ 3 - 3t + 8t - 18 &= 0 \\ 5t - 15 &= 0 \\ t &= 3 \quad \text{(the time of collision)} \end{aligned}$$

Plugging $t = 3$ into the equations for the line yields the point (-2, 12, 9).

Shortcut: If you solve everything symbolically and simplify, you can derive the following expression for the parameter (t) at the intersection of a line and a plane.

$$t = \frac{\mathbf{n} \cdot \langle P_P - P_L \rangle}{\mathbf{n} \cdot \mathbf{v}} \tag{2.31}$$

where $\mathbf{n}$ is the normal to the plane, P_P is a point on the plane, P_L is a point on the line, and $\mathbf{v}$ is the direction vector for the line. If the line and the plane are parallel then $\mathbf{n} \cdot \mathbf{v} = 0$ and there is no intersection unless the line is in the plane, in which case $\mathbf{n} \cdot \langle P_P - P_L \rangle$ is also zero. If there is an intersection point, it is given by

$$\text{Point of Collision} = P_L + t\,\mathbf{v}. \tag{2.32}$$

There is some abuse of notation here because technically speaking you can't add a point to a vector but we allow it here for convenience. This is only a shortcut if you are using a machine to perform the calculations.

- **Example:** Use equations (2.31) and (2.32) to find the point of intersection from the previous example.

Answer:
Here, $\mathbf{n} = \langle 3,\ 2,\ -2 \rangle$, $\mathbf{v} = \langle -1,\ 4,\ 0 \rangle$, $P_P = (0,\ 0,\ 0)$, $P_L = (1,\ 0,\ 9)$, and $\langle P_P - P_L \rangle = \langle -1,\ 0,\ -9 \rangle$. Equation (2.31 yields

$$t = \frac{\mathbf{n} \cdot \langle P_P - P_L \rangle}{\mathbf{n} \cdot \mathbf{v}} = \frac{\langle 3,\ 2,\ -2 \rangle \cdot \langle -1,\ 0,\ -9 \rangle}{\langle 3,\ 2,\ -2 \rangle \cdot \langle -1,\ 4,\ 0 \rangle} = \frac{15}{5} = 3$$

and equation (2.32) yields

$$\text{Point of Collision} = P_L + t\,\mathbf{v} = (1,\ 0,\ 9) + 3\,\langle -1,\ 4,\ 0 \rangle = (-2,\ 12,\ 9).$$

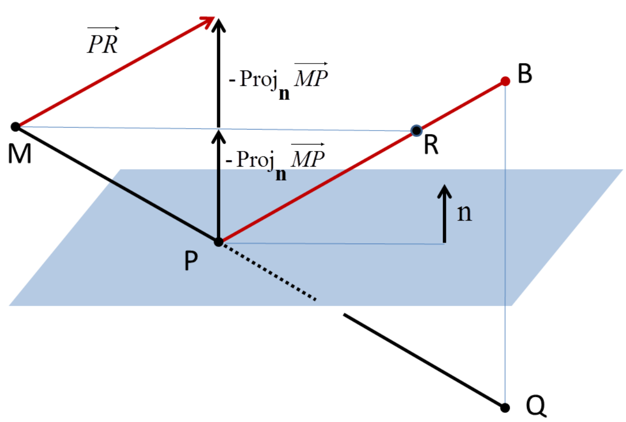
A Bounce Response

Consider an object moving with a velocity vector from point M to point Q. However, it hits a plane at point P. How would the bounce response be modeled?

Some Definitions and Assumptions:

- The collision point is P. You will have to determine the point of intersection as described on the previous page.
- The normal to the plane is given by $\mathbf{n}$ **.
- Let B represent the final resting point after the ball has bounced.
- We will assume the ball travels the same distance after the bounce as it would if there was no collision. That is, there is no energy lost on the bounce and $||\overrightarrow{PQ}|| = ||\overrightarrow{PB}||$.
- We will allow the addition of points and vectors. This addition will result in a point. For example, if P is the point (x, y, z) and $\mathbf{v}$ is the vector $\langle v_1, v_2, v_3\rangle$, then $P + \mathbf{v}$ is the point given by

$$P + \mathbf{v} = (x,\ y,\ z) + \langle v_1,\ v_2,\ v_3\rangle = (x + v_1,\ y + v_2,\ z + v_2)$$



$$\overrightarrow{PR} = \overrightarrow{MP} - 2\ \text{Proj}_{\mathbf{n}}\overrightarrow{MP} \tag{2.33}$$

$$\overrightarrow{PB} = \frac{||\overrightarrow{PQ}||}{||\overrightarrow{PR}||}\ \overrightarrow{PR} \tag{2.34}$$

$$B = P + \overrightarrow{PB} \tag{2.35}$$

** **Note:** Since $\text{Proj}_{\mathbf{n}}\overrightarrow{MP} = \text{Proj}_{(-\mathbf{n})}\overrightarrow{MP}$ it doesn't matter which normal to the plane we use in the formula. In fact, any multiple of any normal will work.

- **Example - A Bounce Response:** Consider the plane defined by $x + z - 2 = 0$. A ball starts at the point $M(-6, 0, 15)$ with a velocity vector that ends at the point $Q(0, 0, -5)$. Find where the ball hits the plane (P) and the final location of the ball (B) after the bounce.

  Answer: First, the normal for our plane is $\mathbf{n} = \langle 1, 0, 1\rangle$. Now we have to determine where the directed line segment $\overrightarrow{MQ}$ hits the plane. So we consider the line which contains this directed line segment. Using the point $M(-6, 0, 15)$ and the direction vector given by $\overrightarrow{MQ} = \langle 6, 0, -20\rangle$.

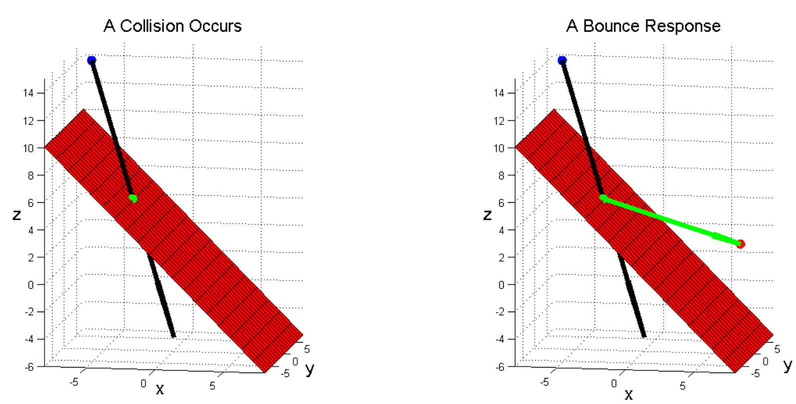
$$\text{Line:} \quad x = -6 + 6t, \quad y = 0, \quad z = 15 - 20t$$

  Substituting the expressions for x, y, and z from the line into the equation for the plane,

$$x + z - 2 = 0 \;\;\rightarrow\;\; (-6 + 6t) + (15 - 20t) - 2 = 0 \;\;\rightarrow\;\; 7 - 14t = 0 \;\;\rightarrow\;\; t = \frac{1}{2}.$$

  The line hits the plane where $t = \frac{1}{2}$ and the point of intersection is $P(-3,\ 0,\ 5)$.
  Now, $\overrightarrow{MP} = \langle 3,\ 0,\ -10\rangle$ and $\overrightarrow{PQ} = \langle 3,\ 0,\ -10\rangle$. It is a coincidence that these two vectors are equal. Now, use equations (2.33) - (2.35) to get the final position of the ball (B).

$$\begin{aligned}
\overrightarrow{PR} &= \overrightarrow{MP} - 2\ \mathrm{Proj}_{\mathbf{n}}\overrightarrow{MP} \\
&= \langle 3,\ 0,\ -10\rangle - 2\ \frac{\langle 3,\ 0,\ -10\rangle \cdot \langle 1,\ 0,\ 1\rangle}{||\ \langle 1,\ 0,\ 1\rangle\ ||^2}\ \langle 1,\ 0,\ 1\rangle \\
&= \langle 3,\ 0,\ -10\rangle - 2\ \frac{-7}{2}\ \langle 1,\ 0,\ 1\rangle = \langle 10,\ 0,\ -3\rangle \\
\overrightarrow{PB} &= \frac{||\overrightarrow{PQ}||}{||\overrightarrow{PR}||}\ \overrightarrow{PR} \\
&= \frac{\sqrt{109}}{\sqrt{109}}\ \langle 10,\ 0,\ -3\rangle = \langle 10,\ 0,\ -3\rangle \\
B &= P + \overrightarrow{PB} = (-3,\ 0,\ 5) + \langle 10,\ 0,\ -3\rangle = (7,\ 0,\ 2)
\end{aligned}$$



See `PlotBounceplane.m` in the Chapter 2 program repository.

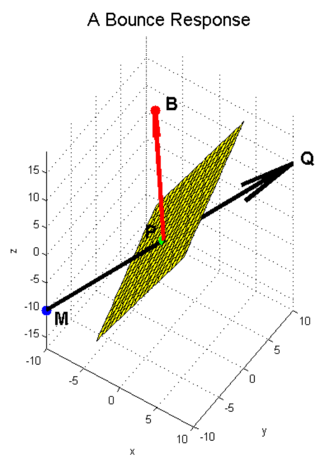
Chapter 2.7 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1.* Consider the plane defined by $2x+y-z+1=0$ and the line connecting the two points $M(-10,-10,-10)$ and $Q(10,10,10)$. Determine where the line intersects the plane. Call this point P.

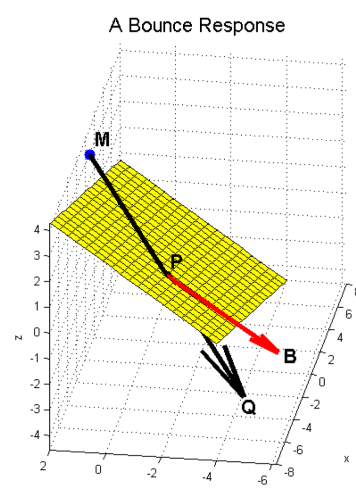
2.* Consider the plane and two points defined in the previous problem. A ball starts at the point M with a velocity vector that ends at Q. It hits the plane at the point found in the previous problem. Determine the final location of the ball when a bounce response to this collision occurs. You can set this up by hand but should have MATLAB® perform the calculations. It gets messy.

3.* **(MATLAB®)** Plot the plane, $\overrightarrow{MQ}$, P, and the final location of the ball (B) after the bounce. It should look something like the graph to the right. You can start with `PlotBouncePlane.m` in the Chapter 2 program repository.



4. Find the point $P(x,y,z)$ where the line connecting the points $M(1,\ 2,\ 3)$ and $Q(-8,\ -5,\ -2)$ intersects the plane defined by $2x-5y+7z-4=0$

5. **(MATLAB®)** Consider the plane and two points defined in the previous problem. A ball starts at the point M with a velocity vector that ends at Q. It hits the plane at P found in the previous problem. Determine the final location of the ball B when a bounce response to this collision occurs. You need to do this in MATLAB® as the calculations get very ugly.
Round your answers to 2 decimal places.

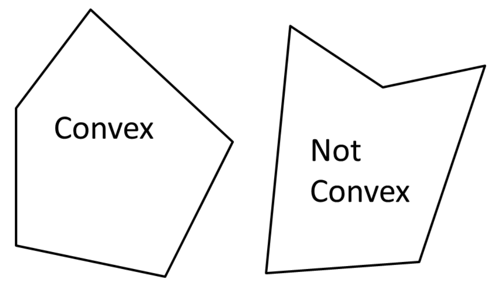


Chapter 2 Project: The Separating Axis Theorem

In the past chapters we have seen a few applications involving collision detection between lines, line-segments, points, and ellipses. This project goes one step further to test for the overlap of any two convex polygons. The following are equivalent definitions of a convex polygon.

Convex Polygons:

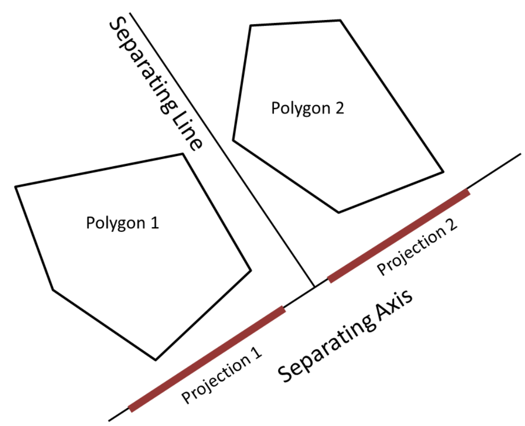
1. Every internal angle is $\leq 180°$.
2. The entire polygon lies entirely on one side of each of its edges.
3. Every line segment between any two vertices remains inside the polygon.
4. There are a few others.



The Separating Axis Theorem (SAT) has some simple forms and some more complicated forms. We'll stick with the two forms in 2D that will be most useful.

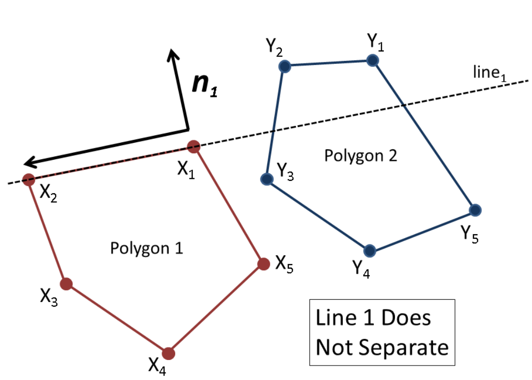
The Separating Axis Theorem (SAT):

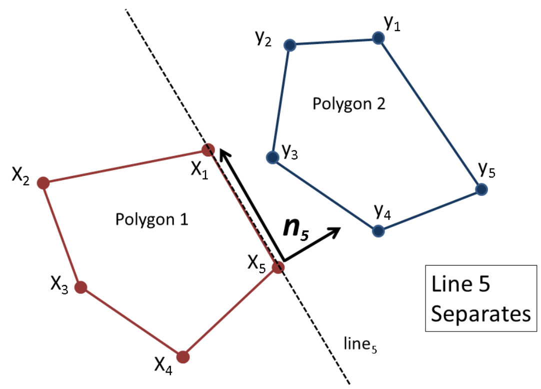
1. **Version 1: Separating Line**
   Two convex polygons do not intersect if and only if there exists a line which separates them.
2. **Version 2: Separating Axis**
   Two convex polygons do not intersect if and only if there exists an axis onto which the projections of the polygons do not overlap.



The SAT implies that if we want to determine whether or not two convex polygons are intersecting, we need to find a separating line or axis. Of course, finding such a line or axis is the tricky part. Fortunately, we only have to check a few candidates. If the polygons do not intersect, there will be at least one separating line parallel to one of the edges. And, for each separating line, there is an associated separating axis perpendicular to it. The strategy is to work your way around the edges of each polygon and check to see if this produces a separating line or axis. Once you find one, you can stop the search. If you go around both polygons and don't find one, the polygons overlap.

Separating Line - Process:



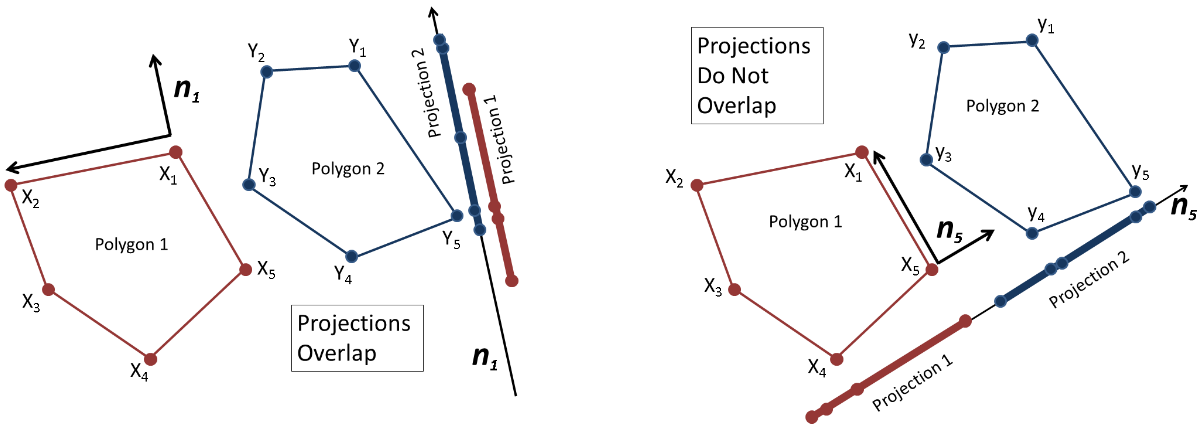


The process is described with respect to the figures above.

- Index the vertices of Polygon 1 as X_i and the vertices of Polygon 2 as Y_i.
- Start with the first edge ($\overrightarrow{X_1X_2}$) on Polygon 1, as seen in the left figure above.
- Rotate $\overrightarrow{X_1X_2}$ by 90° to get $\mathbf{n}_1$ perpendicular to $\overrightarrow{X_1X_2}$.
- Check the sign of $\mathbf{n}_1 \cdot \overrightarrow{X_1X_3}$. This determines whether the angle between $\mathbf{n}_1$ and $\overrightarrow{X_1X_3}$ is acute or obtuse. You don't have to check the other vertices on Polygon 1 because it is convex and the entire polygon lies on one side of every edge. As such the sign of the dot product will not change.
- Check the sign of $\mathbf{n}_1 \cdot \overrightarrow{X_1Y_1}$. It is opposite of $\mathbf{n}_1 \cdot \overrightarrow{X_1X_3}$ (good).
- Check the sign of $\mathbf{n}_1 \cdot \overrightarrow{X_1Y_2}$. It is opposite of $\mathbf{n}_1 \cdot \overrightarrow{X_1X_3}$ (good).
- Check the sign of $\mathbf{n}_1 \cdot \overrightarrow{X_1Y_3}$. It is same as $\mathbf{n}_1 \cdot \overrightarrow{X_1X_3}$ (bad).
  We know the line through X_1 and X_2 does not separate the polygons and you can move to the next edge.
- Keep working through the edges of Polygon 1. When you get to $\overrightarrow{X_5X_1}$ (above right figure), the sign of $\mathbf{n}_5 \cdot \overrightarrow{X_5Y_i}$ is opposite of the sign of $\mathbf{n}_5 \cdot \overrightarrow{X_5X_2}$ for all vertices Y_i on Polygon 2. You have found a separating line.
- If there had been no separating line from Polygon 1, you would do the same process with Polygon 2. Their roles would be reversed.
- If you go around both polygons and don't find a separating line, then the polygons overlap.

Potential Problems: If any 3 sequential vertices of one polygon are collinear, it becomes difficult to tell whether or not you have found a separating line. If the two polygons share an edge or part of an edge, it is also difficult. There are ways to code your way out of these problems but the code gets complicated. If you are worried about these problems, you should use the separating axis routine described on the next page.

Separating Axis - Process:



The process is described with respect to the figures above.

- Index the vertices of Polygon 1 as X_i and the vertices of Polygon 2 as Y_i.
- Start with the first edge ($\overrightarrow{X_1X_2}$) on Polygon 1.
- Rotate $\overrightarrow{X_1X_2}$ by 90° to get $\mathbf{n}_1$ perpendicular to $\overrightarrow{X_1X_2}$ and normalize it by $\mathbf{n}_1 \to \dfrac{\mathbf{n}_1}{||\mathbf{n}_1||}$.
  $\mathbf{n}_1$ will be the axis of concern.
- Note, $\text{Proj}_{\mathbf{n}_1}\overrightarrow{X_1X_i} = \dfrac{\overrightarrow{X_1X_i} \cdot \mathbf{n}_1}{||\mathbf{n}_1||^2}\,\mathbf{n}_1$ for any i.
  Therefore, $\overrightarrow{X_1X_i} \cdot \mathbf{n}_1$ is the signed distance from zero on the $\mathbf{n}_1$ axis.
- Store all values of $\overrightarrow{X_1X_i} \cdot \mathbf{n}_1$ for $i = 2,\ 3, \ldots$ in array 1.
- Store all values of $\overrightarrow{X_1Y_i} \cdot \mathbf{n}_1$ for i $= 1,\ 2,\ 3\ldots$ in array 2.
- See if there is any overlap between the two arrays. There is overlap if
  $$\max(\text{array } 1) \geq \min(\text{array } 2) \qquad \textbf{and} \qquad \min(\text{array } 1) \leq \max(\text{array } 2).$$
- As seen in the left figure above, there is overlap on $\mathbf{n}_1$, so you keep going around Polygon 1.
- As seen in the right figure above, there is no overlap on $\mathbf{n}_5$, so this is a separating axis.
- If there had been no separating axis from Polygon 1, you would do the same process with Polygon 2. Their roles would be reversed.
- If you go around both polygons and don't find a separating axis, then the polygons overlap.

Benefit: This method of collision detection is often used when you also need to calculate a collision response. In this case, the axis with the smallest overlap can be used to help determine the direction of a bounce response.

Assignment:

Go to the Chapter 2 program repositoryand download all of the files. Open and run the file `Project_2_Starter.m`. It calls another program called `Project_2_Setup.m` (in the same folder) that prompts the user for the vertices of both polygons. It plots the polygons and places the vertices into two n x 2 matrices of points.

$$\text{vertices\_1} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{n_1} & y_{n_1} \end{bmatrix} \quad \text{and} \quad \text{vertices\_2} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{n_2} & y_{n_2} \end{bmatrix}$$

The polygons need not have the same number of vertices so the above array of vertices need not have the same number of rows. It also calls a function file called `Convex_Or_Not.m`. This function is used to make sure both polygons are convex. If either of them are not convex, the program exits with an error message. You are to insert the code that uses one of the implementations of the Separating Axis Theorem described on the previous two pages. Be sure to include a variable named `collision`. The value of this variable (0 or 1) will determine the title of the graph.

```
%% Project_2_Starter.m
%  This sets up the polygons for collision detection.
clc; clf; clear;
yes = 1; no = 0;

Project_2_Setup  %This runs the commands in Project_2_Setup.m.
                 %It gets the vertices and plots the polygons.

if Convex_Or_Not(vertices_1) == no        % checks that polygon 1 is convex
    error('first polygon is not convex')  % If not, exit.
end
if Convex_Or_Not(vertices_2) == no        % checks that polygon 2 is convex
    error('second polygon is not convex') % If not, exit.
end

%% You take it from here.
% Include a variable called collision that equals 1 (yes) or 0 (no).

collision = yes;  % replace this line with your code.

if collision == yes
    title('The Polygons Overlap')
else
    title('The Polygons Do Not Overlap')
end
```

Do not edit the files `Project_2_Setup.m` or `Convex_Or_Not.m`. The function file `Convex_Or_Not.m` is depicted on the following page along with some notes. Some of the code in `Convex_Or_Not` can be re-tooled for your code to determine whether or not the polygons overlap.

```
1  function convex = Convex_Or_Not(vertices)
2  %% This function takes in the array of vertices of a polygon.
3  % It returns 1 (yes) if the polygon is convex.
4  % It returns 0 (no) if the polygon is not convex.
5
6  yes = 1; no = 0; % so I can use 'yes' and 'no' instead of 0 and 1
7
8  [n,columns] = size(vertices); % n = # of vertices
9  if (n < 3) | (columns ≠ 2)
10     error('The vertices do not form a polygon')
11 end
12
13 % Below appends another copy of the points to the vertices array.
14 vertices(n+1:2*n , :) = vertices(1:n , :);
15 % This way I can rotate through the edges.
16
17 convex = yes;  %convex unless proven otherwise
18 if n > 3                       % triangles are convex so skip it all.
19     for i = 1:n                % need to check n edges
20         A = vertices(i,:);     % initial vertex of edge
21         B = vertices(i+1,:);   % terminal vertex of edge
22         AB = B-A;              % the vector defining this edge
23         n_vec = [0 , 1; -1, 0] * AB'; % rotate it by 90 degrees.
24         C = vertices(i+2,:);       % next vertex on polygon
25         dp = dot(n_vec,C-A);       % dot product with point
26         current_sign = sign(dp);       % returns 1, -1, or 0.
27         for k = 3:n-1                  % do the remaining vertices.
28             point = vertices(i+k,:);   % next vertex
29             dp = dot(n_vec,point-A);   % dot product with this point
30             new_sign = sign(dp);       % sign of this dot product
31             if current_sign ≠ new_sign  % did dp switch signs?
32                 convex = no;              % if so, not convex
33                 return;                   % and get out of this function
34             end
35         end
36     end
37 end
```

Notes

- This function goes through the edges of the polygon sequentially via the index variable `i`.
- Once it picks an edge from `A` to `B` it then gets a normal vector (`n_vec`) by rotating $\overrightarrow{AB}$ by $90°$.
- It then checks to make sure the angle between that edge and the remaining vertices does not change from acute to obtuse or vice-versa. This is done by checking the sign of the dot product with `n_vec`. If the dot product changes sign, the polygon is not convex.
- To facilitate this process of checking the remaining vertices, I first (line 14) append a copy of the vertices to the end of the initial array of vertices. This way I can just start at vertex `i` and go up to vertex `i + (n-1)` which just circles around the original vertices. You might find this trick useful.
- This function will not recognize a convex polygon if 3 or more consecutive vertices are collinear.

Chapter 3

Vector-Valued Functions

In this chapter we look at vector-valued functions of the form:

2-Dimensional

$$\mathbf{v}(t) = \langle x(t),\ y(t)\rangle \tag{3.1}$$

3-Dimensional

$$\mathbf{v}(t) = \langle x(t),\ y(t),\ z(t)\rangle \tag{3.2}$$

In Chapter 2.5, lines were defined in terms of a set of parametric equations. A vector-valued function is really just a way of expressing a set of parametric equations in vector form. In this section we will take a look at curves traced out by vector-valued functions. In the next section we well differentiate the functions to get velocity and acceleration functions which we well use in describing a projectile in motion. Finally, we will then use Euler's method to determine trajectories based only on the current position and velocity. This last trick is especially handy when animating objects in motion as they bounce off other objects.

3.1 Vector-Valued Functions and Curves

A curve is a one dimensional object within 2D or 3D. These curves can be defined in terms of parametric equations. In 3D, parametric equations look like

$$x = f(t), \quad y = g(t), \quad z = h(t) \tag{3.3}$$

where t is called the *parameter*. In this chapter we will describe these curves equivalently by vector-valued functions of the form

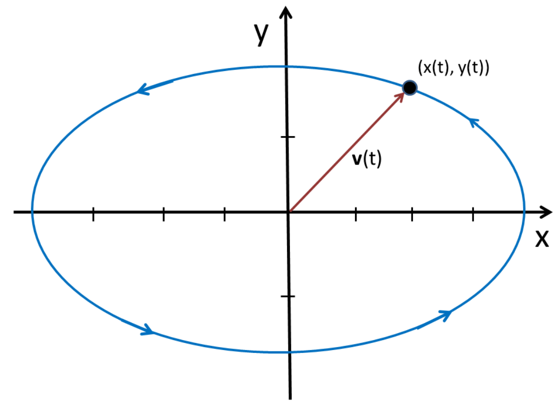
$$\mathbf{v}(t) = \langle x(t),\ y(t),\ z(t)\rangle\,. \tag{3.4}$$

The curves described by vector-valued functions are best visualized by tracing out the terminal point of $\mathbf{v}$ as the parameter t passes through it's domain.

- **Example in 2D - Ellipse:**
  An ellipse in 2D with x radius $= 4$ and y radius $= 2$ can be described by the vector-valued function

  $$\mathbf{v}(t) = \langle 4\cos(t),\ 2\sin(t)\rangle \quad \text{for} \quad t \in [0, 2\pi]$$

  This particular ellipse is traced out once in the counter-clockwise direction starting and ending at (0,4). The direction will become more evident when we look at the associated velocity vectors in the next section.

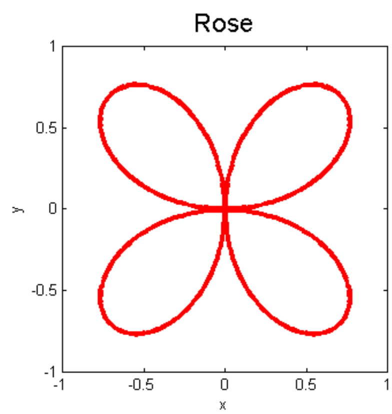


- **Example in 2D - Rose:**
  This rose is traced out by the vector-valued function

  $$\begin{aligned} \mathbf{v}(t) &= \langle \sin(2t)\cos(t),\ \sin(2t)\sin(t)\rangle, \\ t &\in [0, 2\pi] \end{aligned}$$

  The direction is more difficult to ascertain but the curve starts and ends at (0,0) and completes all four *leaves* for $t \in [0, 2\pi]$.

  The MATLAB® code is below.




```matlab
%% This is Rose.m. It plots a Rose.
clc; clf; clear; % Clears command window, figure, and all variables

t = linspace(0,2*pi,300);  % defines the domain with 300 t-values from 0 to 2 pi
x = sin(2*t).*cos(t);      % The x-values. Use .* to multiply term by term.
y = sin(2*t).*sin(t);      % The y-values. Use .* to multiply term by term.
plot(x,y,'r-','linewidth',3);   % plot it with a red line
title('Rose','fontsize',16)     % gives is a title
xlabel('x'); ylabel('y');       % axis labels
grid off; axis equal;           % turn off grid and scale equally
axis([-1,1,-1,1]);              % define the axis bounds [xmin,xmax,ymin,ymax]
```

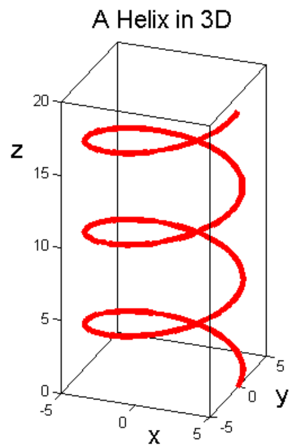
This file (`Rose.m`) is found in the Chapter 3 program repository.

- **Example in 3D - Helix**
  A helix in 3D can be traced out by the vector-valued function

  $\mathbf{r}(t) = \langle 5\cos(t),\ 5\sin(t),\ t\rangle \quad \text{for} \quad t \in [0, 6\pi]$

  This particular helix is traced out in the counter-clockwise direction moving upward (along the z-axis) as t increases. This will become evident when we look at the associated velocity vectors in the next section.

  The MATLAB® code is below.



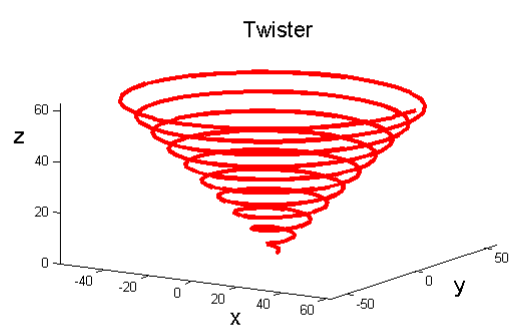

```
%% Helix.m  This plots a helix in 3 Space using the plot3 command
clf; clc; clear; % Clear figure, console, and all variables
%% Start
t = linspace(0,6*pi,200);  % a vector 200 evenly spaced points from 0 to 6 pi.
x = 5*cos(t);              % The corresponding x-values
y = 5*sin(t);              % The corresponding y-values
z = t;                     % The corresponding z-values
plot3(x,y,z,'-r','linewidth',4);       % plot it with a red line
title('A Helix in 3D','fontsize',16);  % the title
axis equal; axis([-5,5,-5,5,0,20])     % [xmin,xmax,ymin,ymax,zmin,zmax]
box on; view(21,24);
xlabel('x','fontsize',16);ylabel('y','fontsize',16);
zlabel('z','fontsize',16);set(get(gca,'ZLabel'),'Rotation',0.0)
```

This file (`Helix.m`) is found in the Chapter 3 program repository.

- **Example in 3D - Twister**
  The *Twister* depicted to the right was traced out by the vector-valued function

$$\begin{aligned} \mathbf{v}(t) &= \langle t\cos(t),\ t\sin(t),\ t\rangle\,, \\ t &\in [0, 20\pi]. \end{aligned}$$

  This curve is traced out in the counter-clockwise direction moving upward (along the z-axis) and outward as t increases.



Converting $y = f(x)$ to a Vector-Valued Function.

In prior math courses you spent a great deal of time defining curves by equations of the form $y = f(x)$. It is very simple to convert these types of functions into vector-valued functions.

If you have the curve defined by

$$y = f(x) \quad \text{for} \quad x \in [a, b]$$

the same curve will be traced by

$$\mathbf{v}(t) = \langle\, t,\ f(t)\, \rangle \quad \text{for} \quad t \in [a, b]$$

- **Example:** Consider the portion of the parabola defined by $y = -x^2 + x + 6$ for $x \in [-2, 3]$. Define the same curve by an equivalent vector-valued function.

  Answer:

  $$\mathbf{v}(t) = \Big\langle\, t,\ -t^2 + t + 6\, \Big\rangle \quad \text{for} \quad t \in [-2, 3].$$

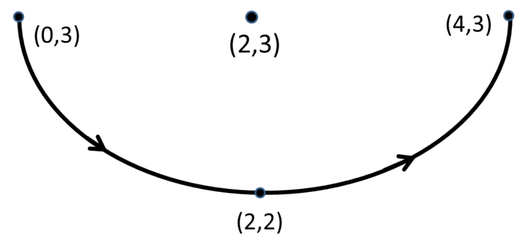
  It is that simple.

Chapter 3.1 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

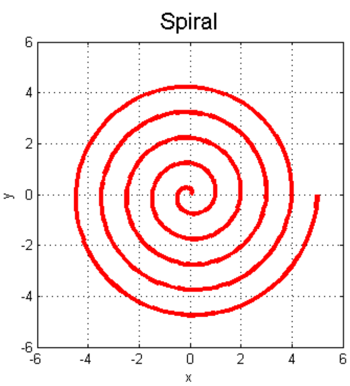
1. **Curves in 2D:**

   (a)* **Lower Half of Ellipse:**
   The vector-valued function $\langle x(t), y(t)\rangle$ traces out the bottom half of an ellipse centered at (2,3) that starts at the point (0,3), moves counter-clockwise through (2,2), and terminates at the point (4,3).

   Give $x(t)$, $y(t)$, and the interval for the parameter t. Then plot it with MATLAB® .

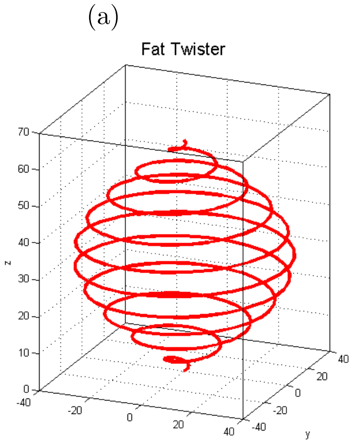
   (b) **Spiral:**
   The vector-valued function $\langle x(t), y(t)\rangle$ traces out the spiral depicted in the figure. The curve should start at (0,0) and go through the points (1,0), (2,0), (3,0), (4,0), and end at (5,0).

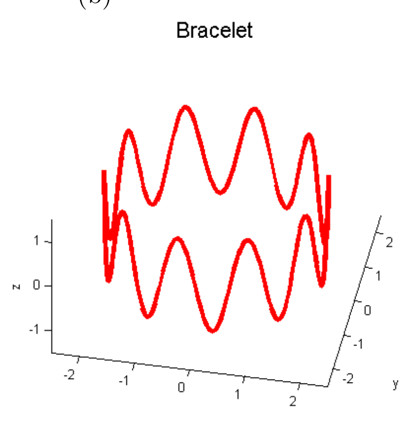
   Give $x(t)$, $y(t)$, and the interval for the parameter t. Then plot it with MATLAB®

2.* **Curves in 3D:** Define a vector-valued function that creates something similar to the graphs depicted below. Use the `plot3` command in MATLAB® to create the graphs.

(a)

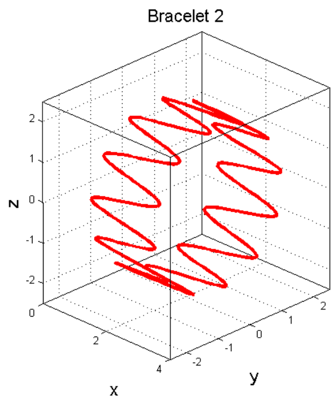


(b)

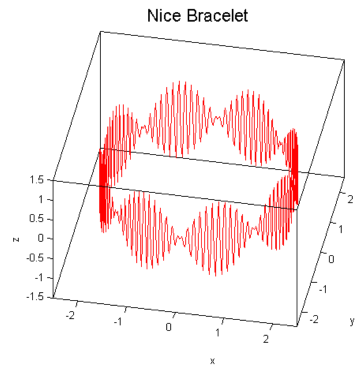


3. **Curves in 3D:** Define a vector-valued function that creates something similar to the graphs depicted below. Use the `plot3` command in MATLAB® to create the graphs.

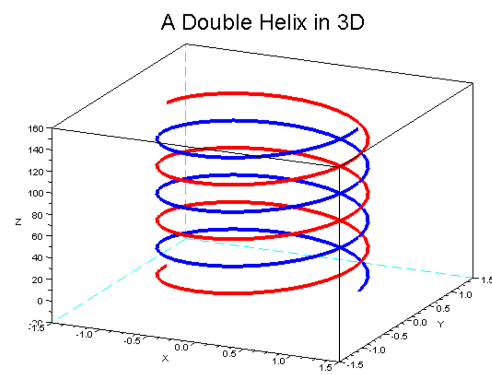
(a)



(b)



4.* **Double Helix:**
Plot a double helix in 3D using the `plot3` function in MATLAB® by making two plots in the same figure. Be sure to use the `hold on;` command after the first plot. It should look something like the one in the figure.



5. **Converting $y = f(x)$ to a vector-valued function:**
For each function $y = f(x)$ in the given domain, convert it to a vector-valued function that generates the same curve and plot it with MATLAB® .

   (a)* The portion of the parabola defined by $y = \frac{1}{10}(x + 20)(20 - x)$ for $x \in [-20, 20]$.

   (b) The portion of the parabola defined by $y = \frac{x}{10}(x - 50)$ for $x \in [0, 50]$.

3.2 Differentiation of Vector-Valued Functions

Differentiation of vector valued functions requires no additional differentiation techniques! You should already know all of the differentiation methods you will need (provided you successfully completed one-variable Calculus). A review of some differentiation rules from Calculus I are found in Appendix B on page 183. The main difference now is the independent variable. In the past it was usually x, now it is usually t.

- One Variable Calculus: Differentiation

$$\text{the derivative of } f(x) = \frac{df}{dx} = f'(x)$$

- Multi-variable Calculus: Differentiation of Vector-Valued-Functions

$$\text{the derivative of } \mathbf{r}(t) = \frac{d\mathbf{r}}{dt} = \mathbf{r}'(t) = \langle x'(t), y'(t), z'(t)\rangle$$

Differentiation: Formal Definition
Technically speaking, the derivative of a vector-valued function $\mathbf{r}(t)$ is given by

$$\mathbf{r}'(t) = \lim_{\Delta t \to 0} \frac{\mathbf{r}(t+\Delta t) - \mathbf{r}(t)}{\Delta t}, \tag{3.5}$$

provided this limit exists. Fortunately, this is exactly the same as you have done in single-variable calculus, only now you do it two or three times (2D or 3D respectively), and you differentiate with respect to the independent variable t instead of x.

Differentiation: Informal Definition
If the vector valued function $\mathbf{r}(t) = \langle x(t),\ y(t),\ z(t)\rangle$, then the derivative of $\mathbf{r}(t)$ is given by

$$\mathbf{r}'(t) = \langle x'(t),\ y'(t),\ z'(t)\rangle \tag{3.6}$$

In words, the derivative of a vector-valued function is just *term by term* differentiation. The resulting vector is one that is parallel to the direction of motion on the curve described by $\mathbf{r}(t)$ as t increases.

Position, Velocity, Speed, and Acceleration: If the original vector-valued function represents the position of a moving object, the position function is usually denoted $\mathbf{r}(t)$ (instead of $\mathbf{v}(t)$) because the velocity vector-valued function is denoted $\mathbf{v}(t)$.

$$\text{Position} = \mathbf{r}(t) = \langle x(t),\ y(t),\ z(t)\rangle\,, \tag{3.7}$$

$$\text{Velocity} = \mathbf{v}(t) = \mathbf{r}'(t) = \langle x'(t),\ y'(t),\ z'(t)\rangle \tag{3.8}$$

$$\text{Speed} = ||\mathbf{v}(t)|| \quad \text{is a non-negative scalar.} \tag{3.9}$$

$$\text{Acceleration} = \mathbf{a}(t) = \mathbf{v}'(t) = \mathbf{r}''(t) = \langle x''(t),\ y''(t),\ z''(t)\rangle \tag{3.10}$$

- **Example in 2D:** Given the position function

$$\mathbf{r}(t) = \langle\, 4\cos(t),\ 2\sin(t)\, \rangle\,, \quad t \in [0, 2\pi]$$

1. Find $\mathbf{v}(t)$ and $\mathbf{a}(t)$

   Answer: Find $\mathbf{v}(t)$ by differentiating $\mathbf{r}(t)$ term-by-term with respect to t

$$\mathbf{v}(t) = \langle -4\sin(t),\ 2\cos(t) \rangle$$

   Differentiating one more time gives acceleration

$$\mathbf{a}(t) = \langle -4\cos(t),\ -2\sin(t) \rangle$$

2. Find the velocity vector and speed at $t = \pi/3$.

   Answer: To find the velocity velocity vector when $t = \pi/3$ we just plug it in;

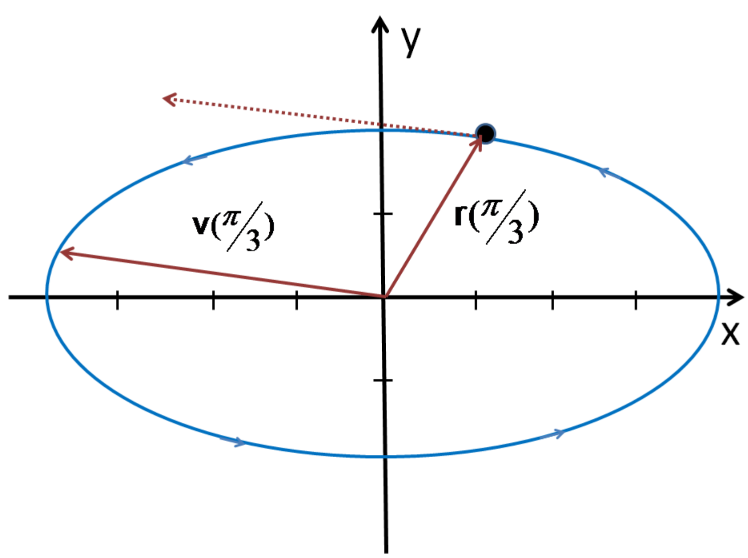
$$\mathbf{v}(\pi/3) = \langle -4\sin(\pi/3),\ 2\cos(\pi/3) \rangle = \left\langle -2\sqrt{3},\ 1 \right\rangle$$

   and the speed is $||\mathbf{v}(\pi/3)||$

$$\text{speed} = \sqrt{(-2\sqrt{3})^2 + 1^2} = \sqrt{13}$$

3. Sketch the graph of $\mathbf{r}(t)$ for $t \in [0, 2\pi]$ along with $\mathbf{r}(\pi/3)$, $\mathbf{v}(\pi/3)$, and $\mathbf{v}(\pi/3)$ originating from the point $\mathbf{r}(\pi/3)$.

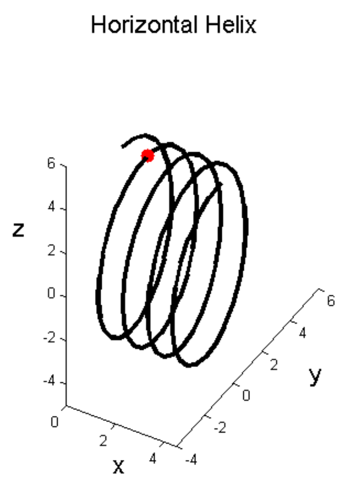
   Answer: $\mathbf{r}(t)$ for $t \in [0, 2\pi]$ is the ellipse traced out in a counter-clockwise direction, $\mathbf{r}(\pi/3)$ and $\mathbf{v}(\pi/3)$ are vectors. And $\mathbf{v}(\pi/3)$ translated to originate at $\mathbf{r}(\pi/3)$ is the dashed vector - it gives the direction of motion at that specific time.

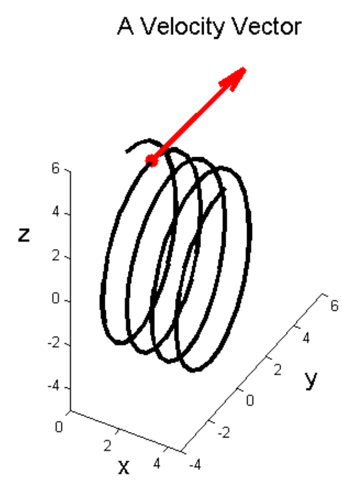


- **Example in 3D:** Consider the helix defined by the vector-valued function:

$$\mathbf{r}(t) = \left\langle \frac{t}{\pi},\ 3\sin(2t),\ 4\cos(2t) \right\rangle \quad \text{for} \quad t \in [0, 4\pi]$$

pictured below (left):





These graphs were created in the file `HorizontalHelix.m` in the program repository.

1. Find $\mathbf{v}(t)$ and $\mathbf{a}(t)$
   Answer: Differentiating term by term,

$$\mathbf{v}(t) = \left\langle \frac{1}{\pi},\ 6\cos(2t),\ -8\sin(2t) \right\rangle \quad \text{and} \quad \mathbf{a}(t) = \langle\ 0,\ -12\sin(2t),\ -16\cos(2t)\ \rangle$$

2. Find the velocity at the point $(1,\ 0,\ 4)$ denoted with a dot in the graph.
   Answer: We really need t, not x, y, and z. However we can figure this out rather easily because if $x = 1$ and $x = t/\pi$, then $t = \pi$. So the velocity vector at this point is the velocity at $t = \pi$, and

$$\mathbf{v}(\pi) = \left\langle \frac{1}{\pi},\ 6\cos(2\pi),\ -8\sin(2\pi) \right\rangle = \left\langle \frac{1}{\pi},\ 6,\ 0 \right\rangle$$

3. Sketch the velocity vector on the curve originating at $(1, 0, 4)$.
   Answer: See graph above (right).

Slope, Velocity, and Tangent Lines

In single-variable calculus you spend a great deal of time getting the slope of a tangent line to the graph of $y = f(x)$. You found the slope of this line to be $y' = f'(x)$. You can easily do the same by differentiating the vector-valued function,

$$\mathbf{r}(t) = \langle t,\ f(t) \rangle \,.$$

- **Example:** Find the slope of the tangent line to the curve $y = x^2$ at the point (3,9).

  1. **Using Single-Variable Calculus:**

     You would consider the function $y = f(x)$ where $f(x) = x^2$.
     Then you would find the derivative of this function denoted $\frac{dy}{dx}$, $\frac{df}{dx}$, $y'(x)$, or $f'(x)$.
     $f'(x) = 2x$, and the slope of the tangent line at $x = 3$ would be $f'(3) = 6$.

  2. **Using Vector Valued Functions:**

     Create the vector-valued function
     $$\mathbf{r}(t) = \left\langle t,\ t^2 \right\rangle$$
     then
     $$\mathbf{v}(t) = \mathbf{r}'(t) = \langle 1,\ 2t \rangle$$
     and the velocity vector at $t = 3$ is given by
     $$\mathbf{v}(3) = \langle 1,\ 6 \rangle \,.$$
     The slope of this vector is $\frac{\text{rise}}{\text{run}} = \frac{6}{1} = 6$.

Summary:

	Single Function		Vector-Valued Function
Function / Position:	$y = f(x)$	$\Longleftrightarrow$	$\mathbf{r}(t) = \langle\ t,\ f(t)\ \rangle$
Slope / Velocity:	slope $= f'(x)$	$\Longleftrightarrow$	$\mathbf{v}(t) = \langle\ 1,\ f'(t)\ \rangle$

Chapter 3.2 Problem Set

Numbers with an asterisk* have solutions in the back of the book.
A review of **differentiation rules** can be found in Appendix B on page 183

1.* Consider the ellipse with x-radius = 2 and y-radius = 8 traced out counter-clockwise by the vector-valued function $\mathbf{r}(t) = \langle\, 2\cos(t),\ 8\sin(t)\rangle\,, \quad \text{for} \quad t \in [0, 2\pi]$.

   (a) Find the velocity function: $\mathbf{v}(t)$.
   (b) Find the velocity vector at the point $P(2, 0)$.
   (c) Find the velocity vector at the point $P(0, -8)$.
   (d) Find the velocity vector at the point $P(-\sqrt{2},\ 4\sqrt{2})$.
   (e) Sketch the ellipse, place the above points on the ellipse, and sketch the velocity vectors originating at the point on the ellipse.

2. Consider the ellipse with x-radius = 2 and y-radius = 8 traced out clockwise by the vector-valued function $\mathbf{r}(t) = \langle\, 2\cos(t),\ -8\sin(t)\rangle\,, \quad \text{for} \quad t \in [0, 2\pi]$.

   (a) Determine the velocity vector-valued function.
   (b) Find the velocity vector at the point $P(2, 0)$.
   (c) Find the velocity vector at the point $P(0, -8)$.
   (d) Find the velocity vector at the point $P(-\sqrt{2},\ 4\sqrt{2})$.
   (e) Sketch the ellipse, place the above points on the ellipse, and sketch the velocity vectors originating at the point on the ellipse.

3.* Consider the 3-D helix described by $\mathbf{r}(t) = \langle \cos(t), \sin(t), t\rangle \quad \text{for} \quad t \in [0, 4\pi]$

   (a) Find $\mathbf{v}(t)$ and $\mathbf{a}(t)$.
   (b) Comment on the acceleration in the z-direction.
   (c) Use MATLAB® to sketch the Helix with the velocity vectors at $t = \pi/2,\ 2\pi$, and $7\pi/2$.

4. Consider the helix described by $\mathbf{r}(t) = \langle\, 4\sin(t), \quad 4\cos(t), \quad 2\pi - t\,\rangle$ for $\quad t \in [0, 4\pi]$

   (a) Find the general velocity vector-valued function: $\mathbf{v}(t)$.
   (b) Find the velocity vector at the point (0,4,0).
   (c) Find the acceleration vector at the point (0,4,0).
   (d) Sketch the velocity and acceleration vectors (from parts (b) and (c)) on the curve originating from the point (0,4,0).

5.* Consider the vector-valued function defined by $\mathbf{r}(t) = \langle t\cos(t), \sin(4t), 3t^2 \rangle$

(a) Find $\mathbf{v}(t)$
You might look up the product rule if you don't remember it.

(b) Find the initial velocity and speed.

(c) Find the acceleration vector: $\mathbf{a}(t) = \mathbf{r}''(t)$.

6.* Consider the vector-valued function given by

$$\mathbf{r}(t) = \left\langle\ 2t,\ t^2+1,\ \frac{1}{2}t^3\ \right\rangle$$

Find the velocity vector at the point (4,5,4).

7. Consider the vector-valued function given by

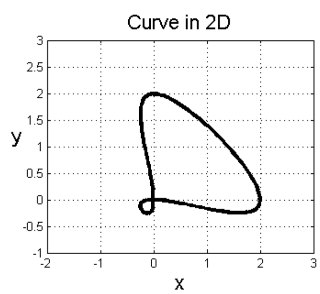
$$\mathbf{r}(t) = \langle\ t\sin(t),\ e^{2t},\ 3t\ \rangle$$

Find the velocity vector at the point $(0,\ e^{2\pi},\ 3\pi)$.

8. Consider the 2D curve sketched to the right and defined by the vector-valued function

$$\mathbf{r}(t) = \langle\ \cos(t)+\cos^2(t),\quad \sin(t)+\sin^2(t)\ \rangle,\quad t \in [0, 2\pi].$$

Determine how this curve is traced as t goes from 0 to 2π by performing the following tasks.



(a) Determine the points on the curve where $t = 0$ and $t = \dfrac{\pi}{2}$.

(b) Find the general velocity vector-valued function $\mathbf{v}(t)$.

(c) Find the velocity vector at the points you found in part (a).

(d) Sketch the velocity vectors from part (c) on the curve. Have them originate from the points you found in part (a). It should now be clear how this curve is traced as t goes from 0 to 2π.

3.3 Projectiles

In this section we will create vector-valued functions to model the motion of a projectile according to Newtonian physics in the absence of air resistance. The equations will be derived with some integration for the 2D model and the formulas transfer quite naturally to the 3D model.

Derivation of the 2D Model

Suppose a projectile is launched with an angle of elevation $= \theta$, an initial velocity (speed) of v_o, starting at a height of h. After the projectile is launched, the only force acting on it is the acceleration due to gravity. We'll let y represent the height of the ball and x represent the horizontal distance from the original launch site and create a vector-valued function for the position of the projectile at all times denoted by $\mathbf{r}(t)$. At time zero, this position vector is $\mathbf{r}(0) = \langle 0,\ h\rangle$. The velocity (speed) at $t = 0$ is v_o with an angle of elevation $= \theta$. With respect to x and y, there are two components of the velocity at time zero, a horizontal component $= v_o\cos(\theta)$ and a vertical component $= v_o\sin(\theta)$. Therefore, the velocity vector at time zero is $\mathbf{v}(0) = \langle v_o\cos(\theta),\ v_o\sin(\theta)\rangle$. Since gravity is pulling it down, we'll represent this force as the constant downward pointing vector $\mathbf{a}(t) = \langle 0,\ -g\rangle$. All of this information is given next the the picture below.

angle of elevation $= \theta$

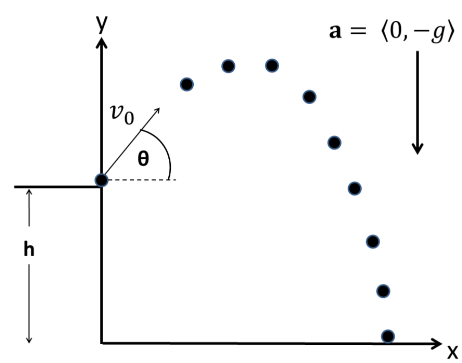
initial velocity (speed) $= v_o$

initial height $= h$

initial position vector: $\mathbf{r}(0) = \langle 0,\ h\rangle$

initial velocity vector: $\mathbf{v}(0) = \langle v_o\cos(\theta),\ v_o\sin(\theta)\rangle$

acceleration vector: $\mathbf{a}(t) = \langle 0,\ -g\rangle$



Notice, we have $\mathbf{a}(t)$ for all time but $\mathbf{r}(t)$ and $\mathbf{v}(t)$ only at time zero. We start with $\mathbf{a}(t)$ and integrate it with respect to time to get $\mathbf{v}(t)$. Integration, like differentiation, is done term-by-term. We use the initial conditions ($t = 0$) to solve for the constants of integration.

$$\begin{aligned}
\mathbf{a}(t) &= \langle 0, -g\rangle \\
\mathbf{v}(t) &= \int \mathbf{a}(t)\ \mathrm{d}t \\
\mathbf{v}(t) &= \langle C_1,\ -gt + C_2\rangle \\
\mathbf{v}(0) &= \langle v_o\cos(\theta),\ v_o\sin(\theta)\rangle \qquad \text{and} \quad C_1 = v_o\cos(\theta) \quad \text{and} \quad C_2 = v_o\sin(\theta) \\
\mathbf{v}(t) &= \langle v_o\cos(\theta),\ -gt + v_o\sin(\theta)\rangle
\end{aligned}$$

Now we have $v(t)$ for all time. This is good. To get the position function we play the same game only now we integrate the velocity function and use the initial conditions on the position to solve for the constants of integration.

$$
\begin{aligned}
\mathbf{r}(t) &= \int \mathbf{v}(t)\ \mathrm{d}t \\
\mathbf{r}(t) &= \int \langle v_o\cos(\theta), \quad -gt + v_o\sin(\theta)\rangle\ \mathrm{d}t \\
\mathbf{r}(t) &= \left\langle [v_o\cos(\theta)]t + C_3, \quad -g\frac{t^2}{2} + [v_o\sin(\theta)]t + C_4 \right\rangle \\
\mathbf{r}(0) &= \langle 0,\ h\rangle \qquad \text{and} \quad C_3 = 0 \quad \text{and} \quad C_4 = h \\
\mathbf{r}(t) &= \left\langle [v_o\cos(\theta)]t, \quad -g\frac{t^2}{2} + [v_o\sin(\theta)]t + h \right\rangle
\end{aligned}
$$

Now we have $\mathbf{r}(t)$ for all time.

A Projectile in Motion - 2D

As derived above we now have a vector-valued function for the position of a projectile according the laws of Newtonian physics in the absence of air resistance in two dimensions. The position of a projectile from an initial height of h with an initial velocity of v_o, and an angle of elevation of θ is given by

$$\mathbf{r}(t) = \left\langle [v_o\cos(\theta)]t, \quad h + [v_o\sin(\theta)]t - \frac{1}{2}gt^2 \right\rangle = \langle x(t),\ y(t)\rangle\,. \tag{3.11}$$

In this equation, g is the acceleration due to gravity = 32 feet per second per second (or 9.8 meters per second per second). To find how far (in the x direction) the ball will travel, you must set $y(t) = 0$, from equation (3.11), and solve for t using the quadratic formula. Simplifying this yields

$$\text{projectile hits the ground at} \quad t^* = \frac{v_o\sin(\theta) + \sqrt{v_o^2\sin^2(\theta) + 2gh}}{g} \quad \text{seconds.} \tag{3.12}$$

The projectile lands at the point $(x(t^*), 0)$. Plugging $t = t^*$ into the equation for $x(t)$ and simplifying yields the total distance traveled in the x-direction given by

$$\text{total distance} \;=\; \frac{v_o^2\cos(\theta)}{g}\left(\sin(\theta) + \sqrt{\sin^2(\theta) + \frac{2gh}{v_o^2}}\right). \tag{3.13}$$

The time t of maximum height is found by differentiating $y(t)$, from equation (3.11), with respect to t, then setting this equal to zero and solving for t. This time will be denoted by $\tilde{t}$.

$$y'(t) = v_o\sin(\theta) - gt = 0 \quad \rightarrow \quad \tilde{t} = \frac{v_o\sin(\theta)}{g} \tag{3.14}$$

Then, to get the actual maximum height, you must insert $\tilde{t}$ into the equation for $y(t)$.

$$\text{maximum height} = h + \frac{v_o^2\sin^2(\theta)}{2g} \tag{3.15}$$

The velocity vector on impact with the ground and speed on impact are given by

$$\text{velocity vector on impact} = \mathbf{v}(t^*) \qquad \text{speed on impact} = ||\mathbf{v}(t^*)|| \tag{3.16}$$

- **Example - Projectile in 2D:** Suppose a football is kicked from the ground at an initial angle of 30^o with an initial velocity of 80 feet per second. Neglect the affect of air resistance. (1) Determine when the ball hits the ground. (2) Determine the vector-valued function that describes the trajectory of the ball. (3) How far will the ball travel in the x direction? (4) How high will it go? (5) What is the velocity vector when it hits the ground? (6) What is the speed on impact with the ground? Give your answers in feet and seconds (use $g = 32$ ft/sec^2).

Answers:

Here, $h = 0$, $\theta = \pi/6$, $v_o = 80$ ft/sec, $g = 32$. Note: $\sin(\pi/6) = \frac{1}{2}$ and $\cos(\pi/6) = \frac{\sqrt{3}}{2}$.

1. Plugging h, θ, v_o, and g into equation (3.12) yields

$$t^* = \frac{v_o \sin(\theta) + \sqrt{v_o^2 \sin^2(\theta) + 2gh}}{g} = \frac{80\frac{1}{2} + \sqrt{80^2 \left(\frac{1}{2}\right)^2}}{32} = 2.5 \text{ seconds}$$

2. Now that we have the total time the ball is in the air, we use equation (3.11) for the actual trajectory:

$$\begin{aligned} \mathbf{r}(t) &= \left\langle [v_o \cos(\theta)]t, \quad h + [v_o \sin(\theta)]t - \frac{1}{2}gt^2 \right\rangle \\ &= \left\langle [80\cos(\pi/6)]t, \quad 0 + [80\sin(\pi/6)]t - \frac{1}{2}32t^2 \right\rangle \\ &= \left\langle \left(80\frac{\sqrt{3}}{2}\right)t, \quad \left(80\frac{1}{2}\right)t - \frac{1}{2}32t^2 \right\rangle \\ &= \left\langle 40\sqrt{3}\,t, \quad 40t - 16t^2 \right\rangle \qquad \text{for} \quad t \in [0, 2.5] \end{aligned}$$

3. To get the total distance traveled we can plug our values into the total distance formula (equation (3.13)) or just plug our ending time $t^* = 2.5$ into $x(t)$ (the first term in $\mathbf{r}(t)$).

$$\text{total disance} = 40\sqrt{3}\,(2.5) = 100\sqrt{3} \approx 173.2 \text{ feet}$$

4. To get the maximum height we plug the same values into equation (3.15) which yields

$$\text{maximum height} = h + \frac{v_o^2 \sin^2(\theta)}{2g} = \frac{80^2 \sin^2(\pi/6)}{64} = 25 \text{ feet.}$$

5. To get the velocity vector when it hits the ground, we plug in $t = t^* = 2.5$ into the velocity vector-valued function.

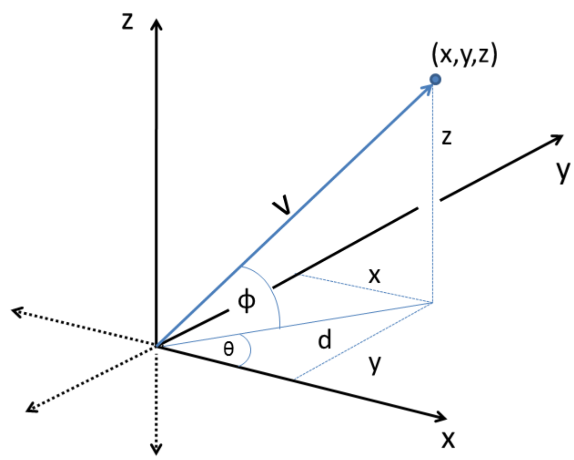
$$\begin{aligned} \mathbf{v}(t) &= \left\langle 40\sqrt{3}, \quad 40 - 32t \right\rangle \\ \mathbf{v}(2.5) &= \left\langle 40\sqrt{3}, \quad 40 - 32(2.5) \right\rangle = \left\langle 40\sqrt{3}, \quad -40 \right\rangle \end{aligned}$$

6. The speed on impact is the magnitude of the velocity vector on impact.

$$\text{speed on impact} = \left\| \left\langle 40\sqrt{3}, \quad -40 \right\rangle \right\| = \sqrt{1600(3) + 1600} = \sqrt{6400} = 80 \text{ ft/sec}$$

Projectile in Motion - 3D Setup

Here we model a projectile in motion in 3D. To start, we now have a couple of new terms: θ is now the angle in the xy-plane made by the initial launch, and ϕ is the elevation angle with respect to the xy-plane.



In this figure, $z = ||\mathbf{v}|| \sin(\phi)$ and $d = ||\mathbf{v}|| \cos(\phi)$. This makes $x = ||\mathbf{v}|| \cos(\phi)\cos(\theta)$ and $y = ||\mathbf{v}|| \cos(\phi)\sin(\theta)$. To summarize:

$$x = ||\mathbf{v}|| \cos(\phi)\cos(\theta) \tag{3.17}$$
$$y = ||\mathbf{v}|| \cos(\phi)\sin(\theta) \tag{3.18}$$
$$z = ||\mathbf{v}|| \sin(\phi) \tag{3.19}$$

The set-up for a projectile in 3D looks very much like that for 2D except

- The variable z denotes the height instead of y.
- The angle of elevation is now ϕ instead of θ.
- The initial velocity (speed), denoted by v_o, has three components given by

$$x\text{-component of initial velocity} = v_o \cos(\phi)\cos(\theta) \tag{3.20}$$
$$y\text{-component of initial velocity} = v_o \cos(\phi)\sin(\theta) \tag{3.21}$$
$$z\text{-component of initial velocity} = v_o \sin(\phi) \tag{3.22}$$

Projectile in Motion - 3D:

The position of a projectile from an initial height h, with initial velocity v_o, an angle with the positive x-axis of θ, and an angle of elevation ϕ, is given by the vector-valued function $\mathbf{r}(t)$ defined by

$$\mathbf{r}(t) = \left\langle v_o \cos(\theta)\cos(\phi)\ t, \quad v_o \sin(\theta)\cos(\phi)\ t, \quad h + v_o \sin(\phi)\ t - \frac{1}{2} g\ t^2 \right\rangle = \langle x(t),\ y(t),\ z(t)\rangle\,. \tag{3.23}$$

In this equation, g is the acceleration due to gravity = 32 feet per second per second (or 9.8 meters per second per second).

To determine when the ball hits the ground, set $z(t) = 0$ and solve for t. This value of t is denoted t^*.

$$\text{projectile hits the ground at} \quad t^* = \frac{v_o \sin(\phi) + \sqrt{v_o^2 \sin^2(\phi) + 2gh}}{g} \quad \text{seconds} \tag{3.24}$$

The ball hits the ground at the point $(x(t^*), y(t^*), 0)$. The distance the ball travels in the xy-plane is

$$\text{total distance in } xy\text{-plane} \ = \ \sqrt{[x(t^*)]^2 + [y(t^*)]^2}. \tag{3.25}$$

The time t of maximum height is found by differentiating $z(t)$, from equation (3.23), with respect to t, then setting this equal to zero and solving for t. This time is denoted by $\tilde{t}$.

$$z'(t) = v_o \sin(\phi) - gt = 0 \quad \rightarrow \quad \tilde{t} = \frac{v_o \sin(\phi)}{g} \tag{3.26}$$

To get the actual maximum height, insert $\tilde{t}$ into the expression for $z(t)$ in equation (3.23):

$$\text{maximum height} = h + \frac{v_o^2 \sin^2(\phi)}{2g} \tag{3.27}$$

The velocity vector on impact with the ground and speed on impact are given by

$$\text{velocity vector on impact} = \mathbf{v}(t^*) \qquad \text{speed on impact} = ||\mathbf{v}(t^*)|| \tag{3.28}$$

Note: These formulas look almost exactly as they do for a projectile in 2D. The only difference is that ϕ now plays the role that θ did, and the total distance traveled is the distance from the landing point in the xy-plane to the origin.

- **Example - Projectile in 3D:** Suppose a projectile is launched from the point (0,0,0). The angle of elevation is 30^o, and the launcher is positioned to point 45^o from the positive x-axis. The initial velocity is 80 feet/second. (1) Determine when it hits the ground. (2) Determine the vector-valued function that describes the trajectory of the ball. (3) Where will it land? (4) How far will it travel in the xy-plane. (5) How high will it go? (6) What is the speed on impact with the ground?

  Answers:

  Here, $h = 0$, $\phi = 30^o = \pi/6$, $\theta = 45^o = \pi/4$, $v_o = 80$ ft/sec, and $g = 32$ feet/sec^2.
  Note: $\sin(\pi/6) = \frac{1}{2}$, $\cos(\pi/6) = \frac{\sqrt{3}}{2}$, and $\sin(\pi/4) = \cos(\pi/4) = \sqrt{2}/2$.

  1. Plugging these values into equation (3.24) for when the ball hits the ground we get
  $$t^* = \frac{v_o \sin(\phi) + \sqrt{v_o^2 \sin^2(\phi) + 2gh}}{g} = \frac{80\frac{1}{2} + \sqrt{80^2\left(\frac{1}{2}\right)^2}}{32} = 2.5 \text{ seconds.}$$
  2. Now that we have the total time the ball is in the air, use equation (3.23) for the trajectory:
  $$\begin{aligned} \mathbf{r}(t) &= \left\langle v_o\cos(\theta)\cos(\phi)t, \quad v_o\sin(\theta)\cos(\phi)t, \quad h + v_o\sin(\phi)t - \frac{1}{2}gt^2 \right\rangle \\ &= \left\langle 80\cos(\pi/4)\cos(\pi/6)t, \quad 80\sin(\pi/4)\cos(\pi/6)t, \quad 0 + 80\sin(\pi/6)t - \frac{1}{2}gt^2 \right\rangle \\ &= \left\langle \left(80\frac{\sqrt{2}}{2}\frac{\sqrt{3}}{2}\right)t, \quad \left(80\frac{\sqrt{2}}{2}\frac{\sqrt{3}}{2}\right)t, \quad \left(80\frac{1}{2}\right)t - \frac{1}{2}32t^2 \right\rangle \\ &= \left\langle 20\sqrt{6}\,t, \quad 20\sqrt{6}\,t, \quad 40t - 16t^2 \right\rangle \qquad \text{for} \quad t \in [0, 2.5] \end{aligned}$$
  3. To find where the ball lands in the xy-plane we just evaluate $x(t)$ and $y(t)$ at $t^* = 2.5$.
  $$x(t^*) = 50\sqrt{6} \approx 122.5, \quad \text{and} \quad y(t^*) = 50\sqrt{6} \approx 122.5$$
  4. To get the total distance traveled in the xy-plane we just determine the distance from
  $$\text{total distance in } xy\text{-plane} = \sqrt{[50\sqrt{6}]^2 + [50\sqrt{6}]^2} = \sqrt{30000} = 100\sqrt{3} \approx 173.2$$
  5. To get the maximum height we plug the same values into equation (3.27) which yields
  $$\text{maximum height} = h + \frac{v_o^2\sin^2(\phi)}{2g} = \frac{80^2\sin^2(\pi/6)}{64} = 25 \text{ feet.}$$
  6. The speed on impact is given by the magnitude of the velocity vector on impact.
  $$\text{speed on impact} = ||\mathbf{v}(2.5)|| = \left|\left|\left\langle 20\sqrt{6}, \quad 20\sqrt{6}, \quad 40 - 32(2.5)\right\rangle\right|\right| = 80 \text{ ft/sec}$$

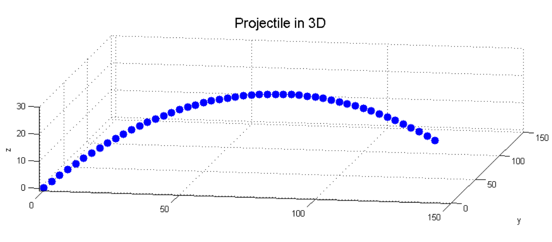
  Note: These answers match those obtained when we did essentially the same problem in 2D.

  The code on the next page does some of these calculations and animates the projectile.

MATLAB® Demonstration - Plotting and Animating Projectiles in 3D

This file (`Projectile3D.m`) does some of the calculations, plots, and animates the projectile described on the previous page. It is available in the Chapter 3 program repository.

```
%% Projectile3D.m
% Plots an animated projectile in the 3D example in Chapter 3.3
clc; clf; clear; %clear command window, figure, and variables

%% Initial Values
vo = 80; thetadegrees = 45; phidegrees = 30; h = 0; g = 32;
theta = thetadegrees*pi/180; phi = phidegrees*pi/180;

%% Plotting directives
xyzbounds = [-1.5, 150, -1.5, 150, -1, 30];  % graph bounds
tsteps = 50;    % number of points to plot

%%    create the trajectory points
tstar = (vo*sin(phi) + sqrt(vo^2 * (sin(phi))^2 + 2*g*h))/g;  %hit time
t = linspace(0,tstar,tsteps);    % the parameter t
x = vo * cos(theta) * cos(phi) * t;     % x(t)
y = vo * sin(theta) * cos(phi) * t;     % y(t)
z = h + vo*sin(phi)*t - 1/2*g*t.^2;      % z(t)

%%  initial plot
trajplot = plot3(x,y,z);  % plot the inital trajectory
hold on; grid on;  %don't erase plot and superimpose a grid
view(10,10); axis equal; axis(xyzbounds);
xlabel('x'); ylabel('y'); zlabel('z');
title('click to fire','fontsize',16);
waitforbuttonpress % waits for a click on the graph.
delete(trajplot);

%%  Animation Loop
for i = 1:1:tsteps
  pointplot = plot3(x(i), y(i), z(i),'b.','markersize',30);
  pause(.05);
end
title('Projectile in 3D','fontsize',16)
```



Projectile in 3D - Velocity and Acceleration

We obtained the position vector for a projectile in motion by first integrating the the acceleration vector to get the velocity vector and then integrated this to get the position vector. Here we go in the other direction by starting with the position vector and differentiate to get the velocity and acceleration vectors. It should come as no surprise that we arrive at the same equations.

The position vector for a projectile in motion is given by equation (3.23),

$$\mathbf{r}(t) = \left\langle v_o \cos(\theta)\cos(\phi)t, \quad v_o \sin(\theta)\cos(\phi)t, \quad h + v_o \sin(\phi)t - \frac{1}{2}gt^2 \right\rangle.$$

where h is the initial height (z-value), v_o is the initial velocity (speed), ϕ is the angle of elevation, θ is the angle made with the positive x-axis, and g is the acceleration due to gravity. This equation can be differentiated with respect to t to get the velocity vector:

$$\mathbf{v}(t) = \langle v_o \cos(\theta)\cos(\phi), \quad v_o \sin(\theta)\cos(\phi), \quad v_o \sin(\phi) - gt \rangle . \tag{3.29}$$

Notice, this is the same velocity vector obtained by integrating the acceleration vector. You can now differentiation this velocity vector to get our original acceleration vector,

$$\mathbf{a}(t) = \langle 0, \ 0, \ -g \rangle . \tag{3.30}$$

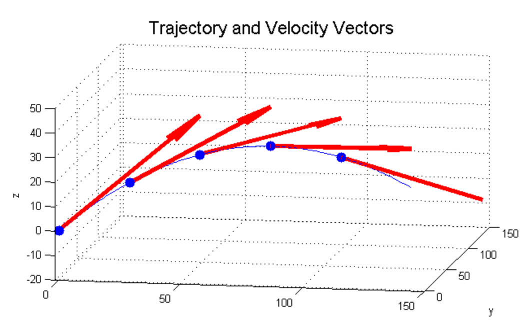
- **Example:** Suppose a projectile is launched from the point (0,0,0). The angle of elevation is 30^o, and the launcher is positioned to point 45^o from the positive x-axis. The initial velocity is 80 feet/second. We found the trajectory for this projectile in the last example using equation (3.23),

$$\mathbf{r}(t) = \left\langle 20\sqrt{6}\, t, \quad 20\sqrt{6}\, t, \quad 40t - 16t^2 \right\rangle .$$

Differentiating this yields the velocity vector,

$$\mathbf{v}(t) = \mathbf{r}'(t) = \left\langle 20\sqrt{6}, \quad 20\sqrt{6}, \quad 40 - 32t \right\rangle .$$

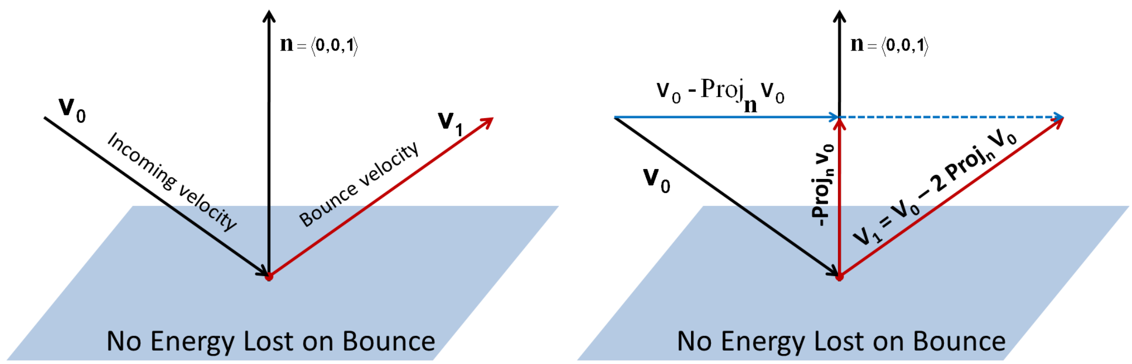
This is a graph of the trajectory with a few velocity vectors superimposed at their associated points of origin. The velocity vector gives the direction of motion while the size of the velocity vector gives the speed. The code used to generate this graph (`Projectile3DWithVelocity.m`) is available from the Chaper 3 program repository



Projectile in 3D - A Bounce off the Ground

Suppose a projectile hits the ground with an incoming velocity vector of $\mathbf{v}_0$. This projectile may then bounce off the ground and we want to know where it will go. We will call the velocity vector after the bounce $\mathbf{v}_1$. This is easily computed once we introduce the normal vector $\mathbf{n} = \langle 0, 0, 1\rangle$. The bounce vector is found using the formula

$$\mathbf{v}_1 = \mathbf{v}_0 - 2\ \mathrm{Proj}_{\mathbf{n}}\mathbf{v}_0 \tag{3.31}$$



Currently, this bounce vector ($\mathbf{v}_1$) only changes the sign of the z component of the incoming velocity vector. This is because the normal vector to the *ground* is $\langle 0, 0, 1\rangle$. Later, when we start dealing with different planes and surfaces, this will not be the case. The formula for $\mathbf{v}_1$ will remain valid but the normal vector will not be $\langle 0, 0, 1\rangle$. Furthermore, it doesn't matter what choice of $\mathbf{n}$ you use, as long as $\mathbf{n}$ is normal to the plane.

Damping: If there is energy lost on the bounce, then the bounce vector will have a smaller magnitude than the incoming vector. This can be modeled with a damping coefficient applied to equation (3.31) as

$$\mathbf{v}_1 = D\ \left(\mathbf{v}_0 - 2\ \mathrm{Proj}_{\mathbf{n}}\mathbf{v}_0\right), \tag{3.32}$$

for $D \in (0, 1]$. If $D = 1$, there is no damping and no energy lost on the bounce. Otherwise,

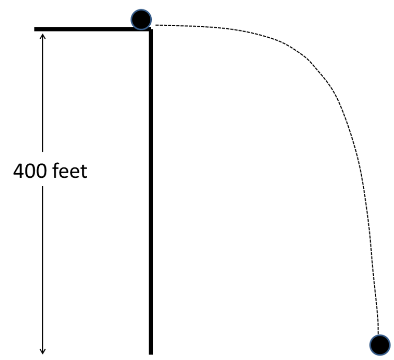
$$||\mathbf{v}_1|| = D\ ||\mathbf{v}_0||.$$

Chapter 3.3 Problem Set

Problems with an asterisk* have solutions in the back of the book.

1.* **Projectile in 2D:**
A ball is kicked from the top of a 400-foot building at an angle of 0^o from horizontal with an initial velocity of 60 feet/second. (see figure).

Let x denote the distance from the base of the building and y denote the height from the ground.



(a) Find the vector-valued function which gives the position (x, y) of the ball. That is, find $\mathbf{r}(t)$.

(b) How high is the ball after 2 seconds?

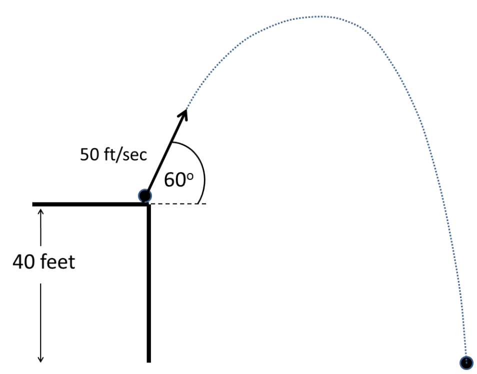
(c) How far from the building does it land?

(d) What is the speed of the ball on impact with the ground?

2. **Projectile in 2D:**
A projectile is launched from the edge of a building 40 feet tall. The initial velocity is 50 feet/second and the angle of elevation is 60^{O}.

Let x denote the distance from the base of the building and y denote the height from the ground.

(figure not necessarily drawn to scale)



(a) Find the vector-valued function which gives the position (x, y) of the ball. That is, find $\mathbf{r}(t)$.

(b) Determine the height (y) when $x = 75$. **Hint:** You will first need to find t when $x(t) = 75$.

3.* **Projectile in 2D:** Consider the projectile in 2D that is launched from an initial height of 28 ft. The angle of elevation is 30^o. The initial velocity is 96 feet/second. Give your answers in feet and seconds (I.e., use $g = 32$ ft/sec^2). Round answers to one decimal place and include units when appropriate.

(a) When will the projectile hit the ground?

(b) What is the vector-valued function for the position of this projectile?

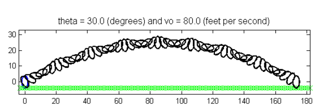
(c) What is the vector-valued function for the velocity of this projectile?

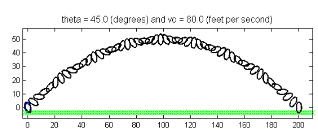
(d) What is speed on impact with the ground?

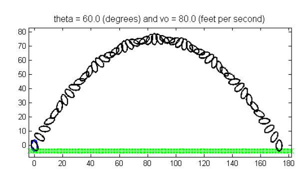
(e) How high will it go?

(f) After launch, how long will it take until it is back at 28 feet high?

4. (**MATLAB® Kick-Off**) Here we visit the kick-off problem from Chapter 1.7 only this time the football should follow a realistic trajectory. Write a program that prompts the user for an initial angle of kick-off (in degrees) and initial velocity (in feet/second). Then have the football follow the trajectory defined by equation (3.11) while rotating in the counter-clockwise direction. Be sure to set your graph bounds so that the trajectory never leaves the graph bounds and takes up most of the graph area. These bounds should be determined by first calculating the total distance using equation (3.13) and the maximum height using equation (3.15). You can force equal scaling with `axis equal` and make MATLAB® obey your graph bounds with `axis([xmin, xmax, ymin, ymax])`. Be sure to impose the bounds **after** setting the axis equal otherwise MATLAB® will change the axis bounds. Sample *kick* animations are displayed below.







5.* **Projectile in 3D:** Consider the projectile that is launched from the point (0,0,60). The angle of elevation is 30^o, and the launcher is positioned to point 45^o from the positive x-axis. The initial velocity is 100 feet/second. Give your answers in feet and seconds (use $g = 32$ ft/sec^2). Include units when appropriate. You are encouraged to use MATLAB® to complete these questions.

(a) Determine when the ball hits the ground. Round to two decimal places.

(b) Determine the vector-valued function that describes the trajectory of the ball.

(c) Where will it land? Round to one decimal place.

(d) How far will it travel in the xy-plane? Round to one decimal place.

(e) How high will it go? Round to one decimal place.

(f) Edit `Projectile3D.m` in the Chapter 3 program repository to plot the trajectory.

6. **Projectile in 3D:** Consider the projectile that is launched from the point (0,0,10). The angle of elevation is 60^o, and the launcher is positioned to point 70^o from the positive x-axis. The initial velocity is 100 feet/second. Give your answers in feet and seconds (use $g = 32$ ft/sec^2). Include units when appropriate. You are encouraged to use MATLAB® to complete these questions.
   (a) Determine when the ball hits the ground. Round to two decimal places.
   (b) Determine the vector-valued function that describes the trajectory of the ball.
   (c) Where will it land?
   (d) How far will it travel in the xy-plane? Round to one decimal place.
   (e) How high will it go? Round to one decimal place.
   (f) Edit `Projectile3D.m` in the Chapter 3 program repository to plot the trajectory.

7.* **Projectile in 3D:** Consider the projectile that is launched from the point (0,0,240). The angle of elevation is 30^o, and the launcher is positioned to point 45^o from the positive x-axis. The initial velocity is 64 feet/second. Give your answers in feet and seconds (use $g = 32$ ft/sec^2). Include units when appropriate. You are encouraged to use MATLAB® to complete these questions.
   (a) Determine when it hits the ground. Give exact answer.
   (b) Where will it land? Round to one decimal place.
   (c) Determine the velocity vector when the projectile hits the ground.
   (d) Determine the speed when it hits the ground. Round to one decimal place.

8.* **Projectile in 3D with Bounce:** Consider the projectile described in problem 5.
   (a) Determine the velocity vector when the ball hits the ground.
   (b) Determine the speed at which the ball is traveling when it hits the ground.
   (c) Determine the velocity vector coming off of the bounce after hitting the ground assuming no energy is lost on the bounce.

9. **Projectile in 3D with Bounce:** Consider the projectile described in problem 6.
   (a) Determine the velocity vector when the ball hits the ground.
   (b) Determine the speed at which the ball is traveling when it hits the ground.
   (c) Determine the velocity vector coming off of the bounce after hitting the ground assuming no energy is lost on the bounce.

10. **Projectile in 3D with Bounce:** Consider the projectile described in problem 7. Determine the velocity vector coming off of the bounce after hitting the ground assuming no energy is lost on the bounce.

3.4 Euler's Method

When dealing with animations of projectiles through space. We will usually only know the current position and the direction in which it is heading. As such, it is impossible to know the full trajectory of the object at all times. So we just go step-by-step. The object is at a current position, it intends on going in a certain direction. But, it might hit something. At this time, we must determine where it will be at the next time step. The way we deal with this *one-step-at-a-time* approach is through a method developed by a guy named Leonard Euler. His last name is pronounced *Oiler.*

Here is one way of describing it mathematically. If

$$\mathbf{r}'(t) = \mathbf{v}(t) = \lim_{\Delta t \to 0} \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t},$$

then if Δt is small we can say that

$$\mathbf{v}(t) \approx \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t},$$

or

$$\mathbf{r}(t + \Delta t) \approx \mathbf{r}(t) + \Delta t\ \mathbf{v}(t). \tag{3.33}$$

If this makes sense to you, then moving points through space is really a simple task. What the above approximation say is this: *if you know the current position of a point ($\mathbf{r}(t)$) and the velocity vector of motion ($\mathbf{v}(t)$), then you can approximate its position as t increases by Δt.* The important part is that the approximation is only valid if Δt is small.

There is some notational issues to resolve here. First, there is the true position vector denoted $\mathbf{r}(t)$. Now, we need to give the approximation a different name. Let's call it $\tilde{\mathbf{r}}(t)$. This notation is a necessary part of the approximation game because we don't want to imply that we know the exact value of $\mathbf{r}(t)$. Assuming we know our original point at $\mathbf{r}(0)$, the approximation sequence looks like this

$$\begin{array}{rcl}
\tilde{\mathbf{r}}(0) & = & \mathbf{r}(0) \quad \text{(known).} \\
\tilde{\mathbf{r}}(\Delta t) & \approx & \tilde{\mathbf{r}}(0) + \Delta t\ \mathbf{v}(0) \\
\tilde{\mathbf{r}}(2\Delta t) & \approx & \tilde{\mathbf{r}}(\Delta t) + \Delta t\ \mathbf{v}(\Delta t) \\
\tilde{\mathbf{r}}(3\Delta t) & \approx & \tilde{\mathbf{r}}(2\Delta t) + \Delta t\ \mathbf{v}(2\ \Delta t) \\
\vdots & = & \vdots \\
\tilde{\mathbf{r}}(n\Delta t) & \approx & \tilde{\mathbf{r}}((n-1)\ \Delta t) + \Delta t\ \mathbf{v}((n-1)\ \Delta t)
\end{array}$$

Now we can continue this process. There are some propagated approximation errors and round-off errors to be expected. However, if you keep Δt small, these errors will be minimized.

Example in 2D: Suppose a ball is kicked from the ground at an initial angle of 30^o with an initial velocity of 80 feet per second. Neglecting the effects of wind or air resistance, depict the ball's approximate trajectory using Euler's method with a time step of $\Delta t = 0.5$ seconds. Note: We know the exact trajectory in this case which is useful for comparing the approximate trajectory.

Answer: We did this problem in Chapter 2.3 and determined the the ball's trajectory was described by

$$\begin{aligned} \mathbf{r}(t) &= \left\langle [v_o \cos(\theta)]t, \quad h + [v_o \sin(\theta)]t - \frac{1}{2}gt^2 \right\rangle \\ &= \left\langle 40\sqrt{3}\, t, \quad 40t - 16t^2 \right\rangle \qquad \text{for} \quad t \in [0, 2.5] \end{aligned}$$

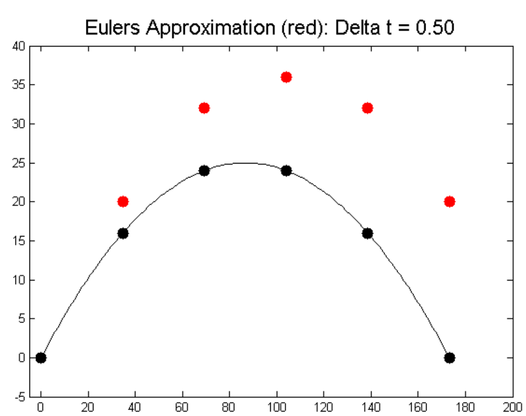
and therefore

$$\begin{aligned} \mathbf{v}(t) &= \langle v_o \cos(\theta), \quad v_o \sin(\theta) - gt \rangle \\ &= \left\langle 40\sqrt{3}, \quad 40 - 32t \right\rangle \end{aligned}$$

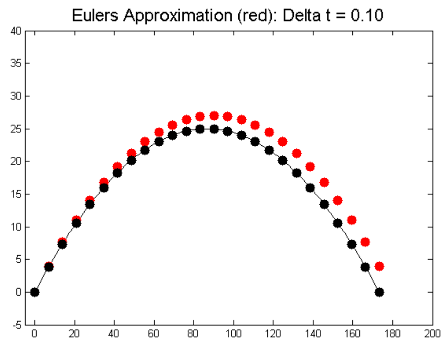
Proceeding through the approximation process with $\Delta t = 0.5$ produces

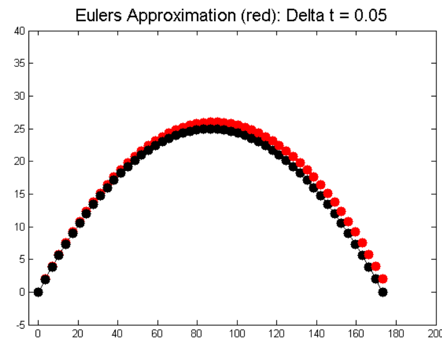
$$\begin{aligned} \tilde{\mathbf{r}}(0) &= (0, 0) \\ \tilde{\mathbf{r}}(.5) &\approx \mathbf{r}(0) + 0.5\,\mathbf{v}(0) = (34.6,\ 20) \\ \tilde{\mathbf{r}}(1) &\approx \tilde{\mathbf{r}}(.5) + 0.5\,\mathbf{v}(.5) = (69.3,\ 32) \\ \tilde{\mathbf{r}}(1.5) &\approx \tilde{\mathbf{r}}(1) + 0.5\,\mathbf{v}(1) = (103.9,\ 36) \\ \tilde{\mathbf{r}}(2.0) &\approx \tilde{\mathbf{r}}(1.5) + 0.5\,\mathbf{v}(1.5) = (138.6,\ 32) \\ \tilde{\mathbf{r}}(2.5) &\approx \tilde{\mathbf{r}}(2.0) + 0.5\,\mathbf{v}(2.0) = (173.2,\ 20) \end{aligned}$$

Below is a depiction of true trajectory (black curve and points) and the Euler approximation (red points above the true trajectory). Notice, this is a very bad approximation.



Euler's method becomes more accurate as the time-step (Δt) gets small





Below is the code used to generate these approximations.
This file (`Euler2D.m`) is in the Chapter 3 program repository.

```
%% This is Euler2D.m in Chapter 3
% This uses Euler's Method to approximate position (R) of a projectile.
clf; clc; clear; % clears console, figure, and all variables respectively

%% Initial Values
Ro = [0,0];     % this is the initial position.
vo = 80; thetadegrees = 30; theta = thetadegrees*pi/180;
tend = 2.5; %final time
Delta_t = 0.1; % time step
tsteps = tend/Delta_t; % number of time steps

%% Plot True Trajectory
tTrue = linspace(0,2.5,100);
x_exact = vo*cos(theta)*tTrue; % exact solution x(t)
y_exact = vo*sin(theta)*tTrue - 16*tTrue.^2; % exact solution y(t)
plot(x_exact,y_exact,'k-');  hold on; % plot the exact trajectory
plot(Ro(1), Ro(2),'k.','markersize',30);  % plot the initial point.
axis([-5,200,-5,40]);

%%  Plot Euler's Approximation
tildeRo = Ro; t = 0;
for i = 1:tsteps
  v = [vo*cos(theta),  vo*sin(theta) - 32*t];         %% current velocity
  tildeR1 = tildeRo + Delta_t*v;                      %% next Euler point
  plot(tildeR1(1),tildeR1(2), 'r.', 'markersize',30)  %% plot it.
  t = t + Delta_t;                                    %% advance time
  R1 = [vo*cos(theta)*t, vo*sin(theta)*t - 16*t^2];   %% True point
  plot(R1(1),R1(2),'k.','markersize',30)              %% Plot it.
  tildeRo = tildeR1;                                  %% Update Euler point
end
text = sprintf('Eulers Approximation (red): Delta t = %1.2f',Delta_t);
title(text,'fontsize',16)
```

The Case of Constant Acceleration

You should notice from the previous development (and code) that we still need to know the velocity function at all times. This might be a problem. However, we can use Euler's Method (again) to approximate velocity. Since acceleration ($\mathbf{a}$) is the derivative of velocity ($\mathbf{v}$),

$$\mathbf{a}(t) = \lim_{\Delta t \to 0} \frac{\mathbf{v}(t+\Delta t) - \mathbf{v}(t)}{\Delta t},$$

then if Δt is small we can say that

$$\mathbf{a}(t) \approx \frac{\mathbf{v}(t+\Delta t) - \mathbf{v}(t)}{\Delta t}, \qquad \text{or} \qquad \mathbf{v}(t+\Delta t) \approx \mathbf{v}(t) + \Delta t\, \mathbf{a}(t).$$

The nice thing here is that if acceleration is constant, this approximation is actually an equality and

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \Delta t\, \mathbf{a} \qquad \text{(provided } \mathbf{a} \text{ is constant)} \tag{3.34}$$

Below is the code that implements this process. Notice, there is no need for any position or velocity function beyond time zero. This file (`Euler2D_R_and_V.m`) is in the Chapter 3 program repository.

```
%% This is Euler2D_R_and_V.m in Chapter 3
% This is a simplified version of Euler2D.
% It uses Euler's Method to approximate position(R)and velocity (V).
clf; clc; clear; % clears command window, figure, and variables
%% Initial Values
vo = 80;                 % Initial Velocity
thetadegrees = 30;       % Initial Angle
h = 0;                   % Initial Height
g = 32;                  % Gravity Constant
tend = 2.5;              % Ending Time
Delta_T = 0.1;           % Step Size
%% Some calculated terms
theta = thetadegrees*pi/180;            % theta in Radians
R = [0,h];                              % Initial Position.
V = [vo*cos(theta), vo*sin(theta)];     % Initial Velocity Vector.
a = [0,-g];                             % Constant Acceleration Vector
tsteps = tend/Delta_T;                  % Number of Time Steps
plot(R(1), R(2),'k.','markersize',30);  % plot the initial point.
hold on; axis([-5,200,-5,40]);
%% Plot the positions
for i = 1:tsteps
  R = R + Delta_T*V;                       % Next Position
  plot(R(1),R(2),'b.', 'markersize',30)    % Plot It.
  V = V+Delta_T*a;                         % Next Velocity Vector.
end
text = sprintf('Eulers Approximation (blue): Delta t = %1.2f',Delta_T);
title(text,'fontsize',16)
```

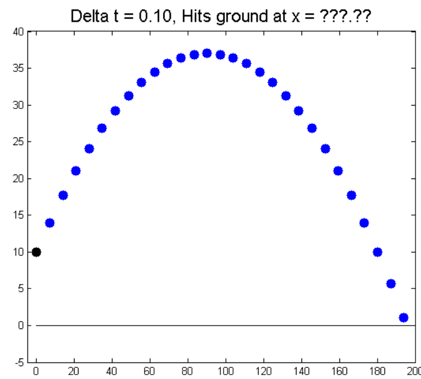
Chapter 3.4 Problem Set

Problems with an asterisk* have solutions in the back of the book.

1. Consider the projectile launched with an initial velocity of 80 ft/sec, at an angle of 30° from an initial height of 10 feet. Assume no air resistance and a gravity constant of 32 ft/(sec)^2.

   (a)* Using the exact solution, determine where the ball hits the ground.

   (b)* Use Euler's Approximation to determine where the ball hits the ground using $\Delta t = 0.10$ seconds. In this case you won't know what the final time is. Just set $\Delta t = 0.10$ and continue to advance time until $\tilde{y} \leq 0$. When this happens, check $\tilde{x}$. That's the approximation to where the ball lands. The graph of the trajectory should look like that in the figure. You can start with `Euler2D.m` or `Euler2D_R_and_V.m` in the Chapter 3 program repository.



   (c) Use Euler's Approximation to determine where the ball hits the ground using $\Delta t = 0.05$ seconds.

   (d) Use Euler's Approximation to determine where the ball hits the ground using $\Delta t = 0.01$ seconds.

   (e)* At what value of Δt does the approximate landing position come within 0.1 feet of the landing position determined in part (a)?

3.5 Bouncing Around in 2D

If you are happy enough with the Euler approximation we can stop worrying about where the object will be for all times and just let time go. This is especially important if you want to bounce around. However, there are some preliminary issues to resolve.

Notation:

$R = (x_o, y_o)$	Current Position (Point)
$\mathbf{v}$	Current Velocity Vector
$\mathbf{V} = \Delta t\ \mathbf{v}$	Change in Position Vector: $\langle \Delta x, \Delta y \rangle$
$R_1 = (x_1, y_1)$	Next Position (Point): $R_1 = R + \mathbf{V}$
$\mathbf{v}_1$	Next Velocity Vector
$\mathbf{V}_1 = \Delta t\ \mathbf{v}_1$	Next Change in Position Vector

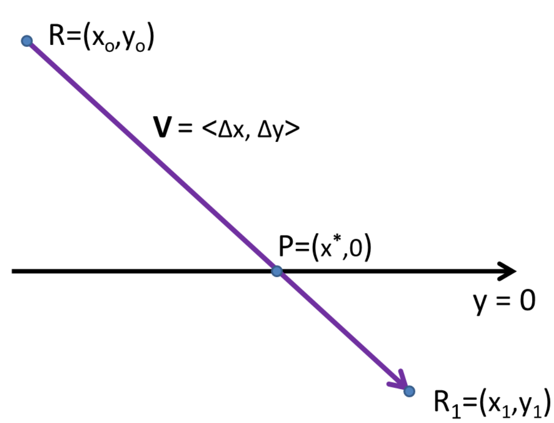
Constant Acceleration: If acceleration vector is constant, we don't need a velocity function for all time. We can use Euler's method to get $\mathbf{v}_1$ from $\mathbf{v}$ by

$$\mathbf{v}_1 = \mathbf{v} + \Delta t\ \mathbf{a} \qquad \text{where} \qquad \mathbf{a} = \langle 0, -g \rangle$$

Bounce Point P:

When bouncing around, it is unlikely the ball will hit the ground at exactly $y = 0$. More than likely, R_1 will be below ground. When this happens, we want to determine where the ball actually hits $y = 0$. The parametric equations for line from R to R_1 are given by $R + s\mathbf{V}$ or

$$x = x_o + s\ \Delta x \qquad \text{and} \qquad y = y_o + s\ \Delta y$$



Denote P by $(x^*, 0)$ and find x^*

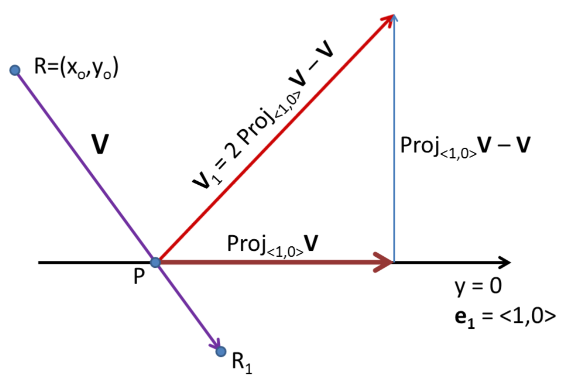
Find s so that $y_o + s\ \Delta y = 0$

to get $s = -\dfrac{y_o}{\Delta y}$

so $x^* = x_o - \dfrac{y_o}{\Delta y}\Delta x$

This could be a problem if $\Delta y = 0$.
But the ball can't go from above $y = 0$ to below $y = 0$ if $\Delta y = 0$.
So this shouldn't happen.

Bounce Vector $\mathbf{V}_1$:



If $\mathbf{V}$ takes the ball below $y = 0$, we want to bounce from P in the appropriate direction. With a *perfect bounce* (no loss of energy), the bounce vector ($\mathbf{V_1}$) will have the same magnitude ($\mathbf{V}$). In this case, the bounce vector $\mathbf{V}_1$ is found by

$$\mathbf{V}_1 = 2\ \text{Proj}_{\langle 1,0\rangle}\mathbf{V} - \mathbf{V} \tag{3.35}$$

Note: We just calculated the change in position vector($\mathbf{V}_1$). The new velocity vector is given by

$$\mathbf{v_1} = \frac{1}{\Delta t}\mathbf{V_1}$$

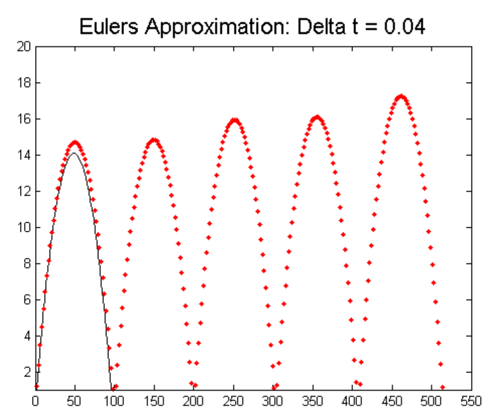
Damping: If there is energy lost on the bounce, then the bounce vector will have a smaller magnitude than the incoming vector. This can be modeled with a damping coefficient applied to equation (3.35) as

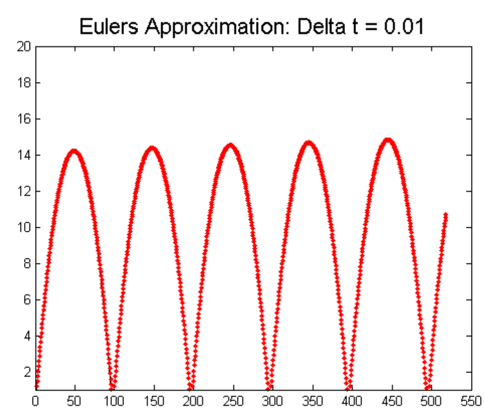
$$\mathbf{V}_1 = D\ \left(2\ \text{Proj}_{\langle 1,0\rangle}\mathbf{V} - \mathbf{V}\right) \tag{3.36}$$

for $D \in (0, 1]$. If $D = 1$, there is no damping and no energy lost on the bounce. Otherwise,

$$||\mathbf{V}_1|| = D\ ||\mathbf{V}||.$$

Let Time Roll: Here is a bouncing ball animation with no damping ($D = 1$).





Notice, the ball always goes a little higher than it should. Therefore it hits the ground a little harder than it should and then bounces a little higher than it should. As such, this type of model (with no damping) will tend to lead to higher and higher bounces. That's why a little damping is always a good idea. The code is on the next page.

MATLAB® code: This code creates the bouncing ball animation on the previous page. The file (`Euler2DBounce`) is in the Chapter 3 program repository.

```
%% This is Euler2D_Bounce.m
% Performs a bouncing ball simulation.
clf; clc; clear;

%% Initial Values
vo = 60; thetadegrees = 30; h=0; g = 32;
t_end = 10; Delta_t = .04; t_steps = t_end/Delta_t;
R=[0,0];   %/ initial position.
theta = thetadegrees*pi/180;
accel = [0,-g];   %/ the constant acceleration vector.
D = 1;   %/ D = damping on the bounce (between 0 and 1)

%% Initial graph up to first bounce.
Initialtend = (vo*sin(theta) + sqrt((vo*sin(theta))^2 + 2*g*h))/g;
t = linspace(0,Initialtend,50);
xtraj = R(1) + (vo * cos(theta))*t;
ytraj = R(2) + vo *sin(theta)*t - (g/2)*t.^2;
plot(xtraj,ytraj,'k-'); hold on; plot(R(1),R(2),'r.');
axis([0,550,1,20]);
title('click to start','fontsize',16);
waitforbuttonpress % waits for a click on the graph.
title('running','fontsize',16);

%% Animation Loop
v = vo*[cos(theta), sin(theta)];   % initial velocity vector
for i = 1:1:t_steps
  V = Delta_t * v;          % Current Position Change
  R1 = R + V;               % Next Position
  v1 = v + Delta_t*accel;   % Next Velocity Vector (no bounce)
  if R1(2)<0                               % Collision Detection: Is y1 < 0?
    disp('bounce')                         % Display 'bounce' to console
    R1 = [R(1) - R(2)* v(1)/v(2), 0];     % Intersection Point
    V1 = D*(2*ProjUV(V,[1,0]) - V);        % Next Change in Position Vector
    v1 = V1/Delta_t;                       % Next Velocity Vector
  end
  pause(.005);
  plot(R1(1),R1(2), 'r.'); hold on; %pause(.01)  %% plot the new point.
  R = R1; v = v1;   % Reset R and V to the updated values.
end
text = sprintf('Eulers Approximation: Delta t = %1.2f',Delta_t);
title(text,'fontsize',16)
```

Bouncing Off Other Things: In the previous example we bounced a ball off of the $y = 0$ axis. Here we look at finding the bounce point and bounce vector when a ball bounces off some other line. Suppose $R(x_o,\ y_o)$ represents the original point of the ball and the change in position vector ($\mathbf{V}$) has it landing in the other side of a wall at R_1. The wall is defined by an original point $W(x_w,\ y_w)$ and a terminal point W_1. The wall vector is denoted $\mathbf{w} = \overrightarrow{WW_1}$.

Bounce Point P: We want to find P

The line from R to R_1 is $R + s\ \mathbf{V}$

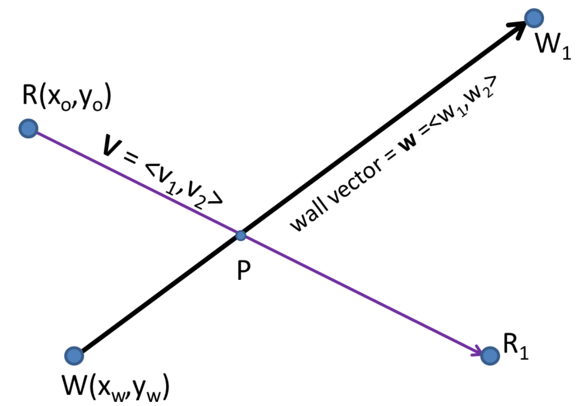
$$x = x_o + s\ v_1$$
$$y = y_o + s\ v_2$$

The line from W to W_1 is $W + r\ \mathbf{w}$

$$x = x_w + r\ w_1$$
$$y = y_w + r\ w_2$$

Goal: Find s and r so that

$$x_o + s\ v_1 = x_w + r\ w_1$$
$$y_o + s\ v_2 = y_w + r\ w_2$$



$$\begin{array}{rcl} x_o + s\ v_1 & = & x_w + r\ w_1 \\ y_o + s\ v_2 & = & y_w + r\ w_2 \end{array} \sim \begin{array}{rcl} -r\ w_1 + s\ v_1 & = & x_w - x_o \\ -r\ w_2 + s\ v_2 & = & y_w - y_o \end{array} \rightarrow \begin{bmatrix} -w_1 & v_1 \\ -w_2 & v_2 \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} x_w - x_o \\ y_w - y_o \end{bmatrix}$$

Notice, the determinant of the matrix is $-w_1v_2 + w_2v_2$ which is only zero if $\mathbf{w}$ is parallel to $\mathbf{v}$. In this case, there should be no intersection and no intersection point. Solving this matrix equation gets us s and r, either of which can be used to get the bounce point by

$$P = R + s\ \mathbf{V} \qquad \text{or} \qquad P = W + r\ \mathbf{w}$$

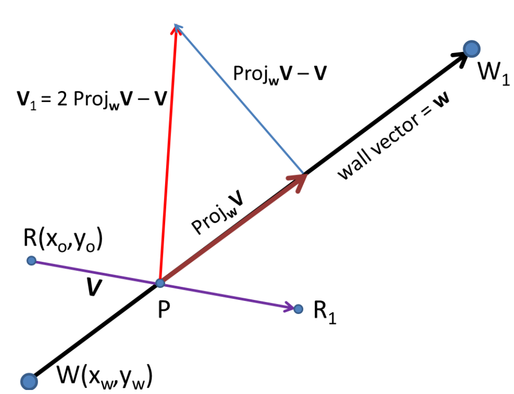
Bounce Vector $\mathbf{V}_1$:

The bounce vector is found by the exact same procedure used to find the bounce vector off the ground. The only difference is that the ground-vector $\langle 1, 0\rangle$ is replaced with the wall-vector $\mathbf{w}$.

$$\mathbf{V}_1 = 2\ \mathrm{Proj}_{\mathbf{w}}\mathbf{V} - \mathbf{V} \tag{3.37}$$

and

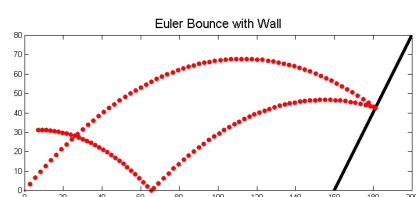
$$\mathbf{v_1} = \frac{1}{\Delta t}\mathbf{V_1}$$



<u>MATLAB® code:</u> `Euler_2D_Bounce_Wall.m` is in the Chapter 3 program repository. It performs Euler's method for a projectile with a bounce on a non-horizontal wall defined by a line segment through two points.

```
%% This is Euler_2D_Bounce_Wall.m
% Performs a bouncing ball simulation with a non-horizontal wall
clf; clc; clear;
%% Initial Values
vo = 85; thetadegrees = 50; h=0; g = 32;
t_end = 7; Delta_t = .05; t_steps = t_end/Delta_t;
R=[0,h];    %/ initial position.
theta = thetadegrees*pi/180;
accel = [0,-g];   %/ the constant acceleration vector.
D = .8;    %/ D = damping on the bounce (between 0 and 1)
%% Wall Points (W0 & W1), Wall Vector (w), & Wall Slope (m)
W0=[160,0]; W1=[200,80]; w = W1-W0; m = (W1(2)-W0(2))/(W1(1)-W0(1));
%% Plot first Point and  Wall
plot(R(1),R(2),'r.'); hold on;
plot([W0(1),W1(1)],[W0(2),W1(2)],'k-','linewidth',4);
hold on; axis equal; axis([0,200,0,80]);
title('click to start','fontsize',16);
waitforbuttonpress % waits for a click on the graph.
title('Euler Bounce with Wall','fontsize',16);
%% Animation Loop
v = vo*[cos(theta), sin(theta)];   % initial velocity vector
for i = 1:1:t_steps
  V = Delta_t*v;
  R1 = R + V;        % Next Position
  v1 = v + Delta_t*accel;   % Next Velocity Vector
  if R1(2)<0                              % Detect ground hit
      disp('hit ground');
      V1 = D*(2*ProjUV(V,[1,0]) - V);     % New V1
      R1 = [R(1) - R(2)* V(1)/V(2), 0];   % [x*, 0] = New R1
      v1 = V1/Delta_t;                    % New v1
  end
  if R1(1) ≥ W0(1) & R1(1) ≤ W1(1) & R1(2) ≤ W0(2)+ m*(R1(1)-W0(1))
      disp('hit wall')                    % detect wall hit**
      V1 = D*(2*ProjUV(V,w) - V);         % New V1
      rs=[-w(1),V(1);-w(2),V(2)]\[W0(1)- R(1); W0(2) - R(2)]; % Collision r & s
      R1 = R + rs(2)*V;                   % Collision Point = New R1
      v1 = V1/Delta_t;                    % New v1
  end
  pause(.005);
  plot(R1(1),R1(2), 'r.','markersize',20);   %% plot the new point.
  R = R1; v = v1;   % Reset R and v.
end
```

**Collision with Wall Notes:
The wall is defined by the points W0(x_0, y_0) and W1(x_1, y_1). The line is defined by $y - y_0 = m(x - x_0)$, where m is the slope of the wall. The ball R1(x,y) hits or crosses the wall if $x \in [x_0, x_1]$ and $y \leq y_0 + m(x - x_0)$.



Chapter 3.5 Problem Set

Problems with an asterisk* have solutions in the back of the book.

1. Edit the MATLAB® file `Euler2Dbounce.m` to simulate a bouncing ball with the following requirements:

   (a) The ball starts at (0,20).

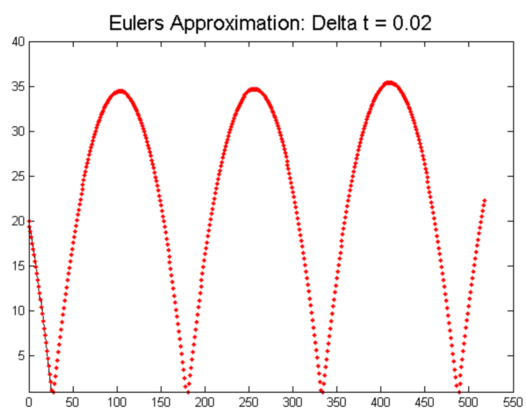
   (b) The ball is launched with an initial velocity of 60 feet per second.

   (c) The angle of elevation is -30^o.

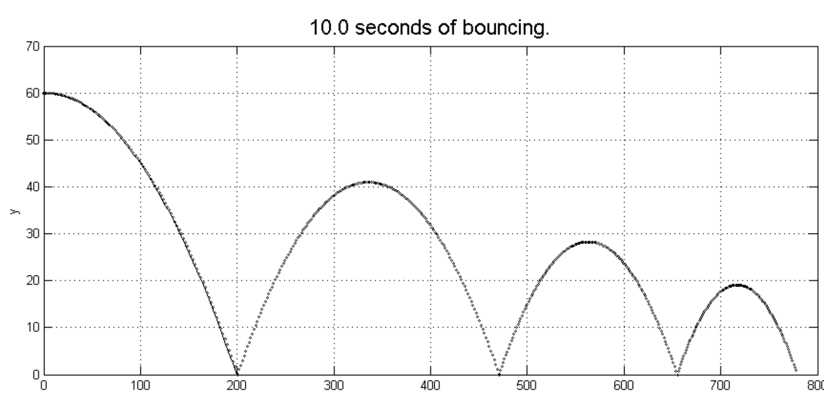
   (d) The simulation runs for 10 seconds.

   (e) The time step is 0.02 seconds.

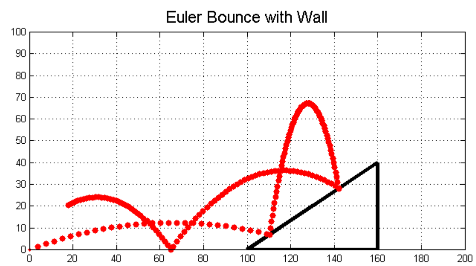
   Your final graph should look like the the one below.



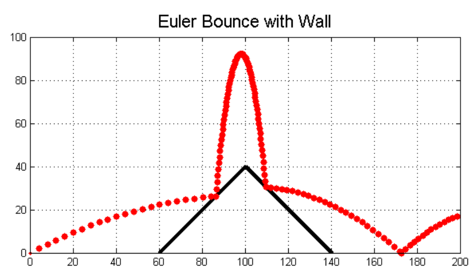
2. Edit the MATLAB® file `Euler_2D_bounce_Wall.m` to simulate a bouncing ball that results in the following trajectory over the course of 10 seconds. You should be able to match all the bounce points and bounce heights. Let $\Delta t = 0.02$.



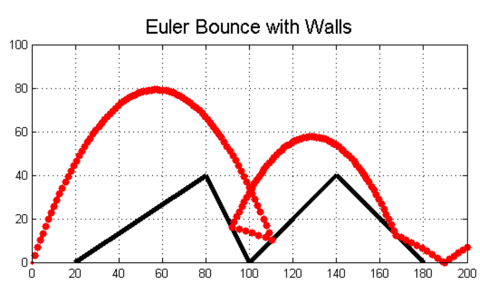
3.* Edit the file `Euler_2D_bounce_Wall.m` to simulate a bouncing ball depicted below.
Program Parameters : Total time = 9 seconds, $v_o = 80$, $\theta = 20°$, $D = 0.8$, $\Delta t = 0.05$
Vertices of the Triangle: (100,0), (160,40), and (160,0).



4. Edit the file `Euler_2D_bounce_Wall.m` to simulate a bouncing ball depicted below.
Program Parameters : Total time = 9 seconds, $v_o = 90$, $\theta = 27°$, $D = 0.8$, $\Delta t = 0.05$
Vertices of the Triangle: (60,0), (100,40), and (140,0).



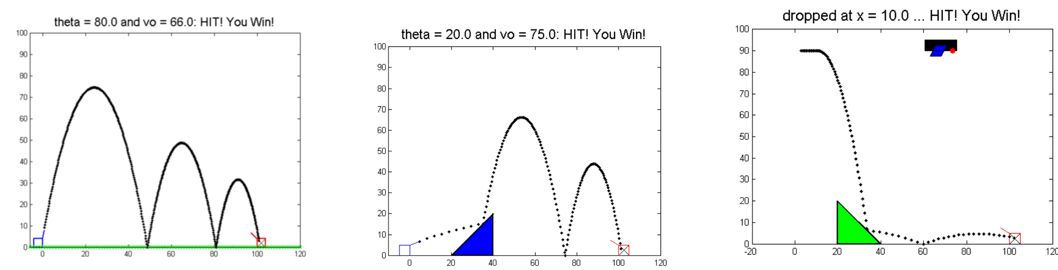
5. Edit the file `Euler_2D_bounce_Wall.m` to simulate a bouncing ball depicted below.
Program Parameters: Total time = 10 seconds, $v_o = 75$, $\theta = 70°$, $D = 0.9$, $\Delta t = 0.05$
Vertices of the Triangles: (20,0), (80,40), (100,0), (140,40) and (180,0)



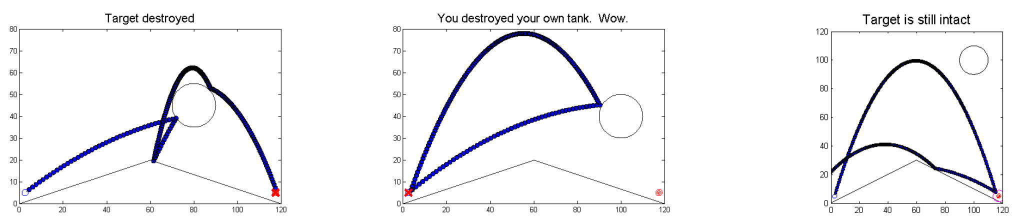
Chapter 3 Project: Projectile Game in 2D

Create some type of 2D game with a projectile and a target. The projectile must follow a trajectory described by Newtonian Physics as estimated by Euler's Method. There should be some type of bouncing and some type of collision detection to determine whether or not a target has been hit. This type of game often takes the form of a tank shooting at a target. You are free to change this theme to one of your choosing. Be creative. Below are screenshots of some examples.

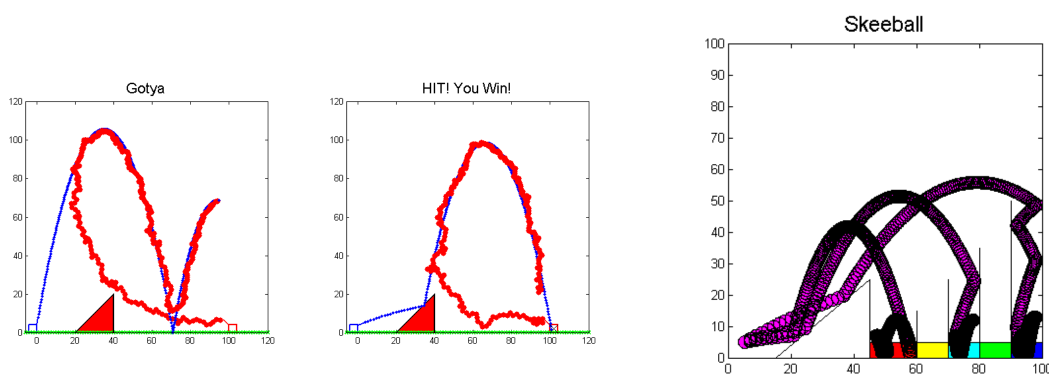
- Tank game with bouncing, a ramp, and an airplane drop.



- Tank game where a *bumper* is hit to disable the target shield. You can also destroy your own tank.



- Tank game with air defense system and a Skeeball game.



Chapter 4

Multi-Variable Functions and Surfaces

A surface in 3D is created by any equation of the general form:

$$F(x, y, z) = 0 \tag{4.1}$$

Here, $F(x, y, z)$ is a **multi-variable function** and the surface is defined **implicitly**. Usually, but not always, the same surface can be expressed in one of these three **explicit** forms:

$$z = f(x, y), \qquad y = g(x, z), \quad \text{or} \quad x = h(y, z). \tag{4.2}$$

In these cases, f, g, and h are multi-variable functions with two variables instead of three. We will focus on explicit functions of the first form in (4.2).

4.1 Surfaces: $z = f(x, y)$

Here we consider multi-variable functions of the form

$$z = f(x, y) \tag{4.3}$$

The function here, $f(x, y)$, is the multi-variable function. There are two independent variables, x and y, and one dependent variable; z.

A Familiar Surface - Planes

We have already seen one example of a surface defined implicitly in 3-space.

$$\text{plane:} \quad ax + by + cz - d = 0$$

Provided $c \neq 0$, we can define this surface explicitly as

$$\text{plane:} \quad z = \frac{-ax - by + d}{c} = -\frac{a}{c}\,x - \frac{b}{c}\,y + \frac{d}{c} = f(x, y) \tag{4.4}$$

If $c = 0$ we will have to choose one of the other explicit forms from (4.2). Notice, this looks a lot like the equation for a line in 2D. Here, d/c is z-intercept. The *slope* is defined in terms of the coefficients of x and y. We will resolve the issue of slope later in this chapter.

Plotting Traces and Level Curves

Consider the surface created by the multi-variable function:

$$z = x^2 + y^2 + 2 \tag{4.5}$$

To sketch something like this by hand it is best to start by sketching **traces** in various planes.

- The **trace** in the xz-plane is found by letting $y = 0$.

  $$z = x^2 + 2$$

  which is just a parabola in the xz-plane.

- The **trace** in the yz-plane is found by letting $x = 0$.

  $$z = y^2 + 2$$

  which is another parabola in the yz-plane.

- The **trace** in the xy-plane is found by letting $z = 0$.

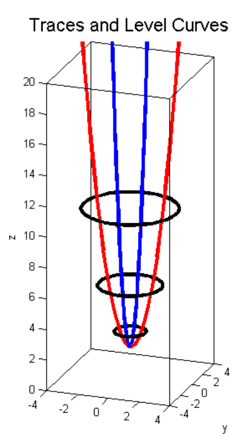
  $$0 = x^2 + y^2 + 2 \quad \text{or} \quad x^2 + y^2 = -2$$

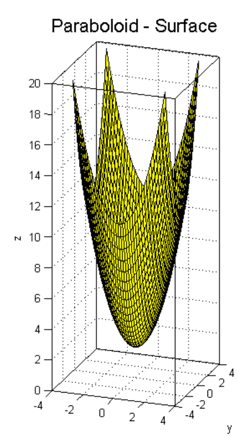
  and this has no solutions. As such, there is no *trace* in the xy-plane.

- Try some traces at different values of z. These are called **level curves**.

  1. $z = 2 \quad \rightarrow \quad x^2 + y^2 = 0$. This is just the point $(x, y) = (0, 0)$.
  2. $z = 3 \quad \rightarrow \quad x^2 + y^2 = 1$. This is the circle centered at (0,0) with radius 1.
  3. $z = 6 \quad \rightarrow \quad x^2 + y^2 = 4$. This is the circle centered at (0,0) with radius 2.
  4. $z = 11 \quad \rightarrow \quad x^2 + y^2 = 9$. This is the circle centered at (0,0) with radius 3.

When you piece all of the traces and level curves together you should get the picture of the surface which is called a paraboloid.





The code used to create these graphs is on the next page

MATLAB® Code: `Plotting_Paraboloid_Traces.m` in the Chapter 4 program repository.

```
%% This is Plotting_Paraboloid_Traces.m
% It plots some traces for z = x^2 + y^2 + 2 with the param3d function.
clc; clf; clear; % clears command window, figure, and variables

%% The function z = f(x,y) = x^2 + y^2 + 2
z = @(x,y)[x.^2 + y.^2 + 2];

%%  Plot trace in the xz-plane
xvec = linspace(-4,4,40);  % create the vector of x-values
yvec = linspace(-4,4,40);   % create the vector of y-value

%% Plot the two traces using the plot3 command
plot3(xvec, 0*yvec, z(xvec,0*yvec),'r-','linewidth',3); hold on;
plot3(0*xvec, yvec, z(0*xvec,yvec),'b-','linewidth',3)
xlabel('x'); ylabel('y'); zlabel('z'); box on;
view(20,20); axis equal; axis([-4,4,-4,4,0,20]);

%% Plot the three level curves using the plot3 command
tvec = linspace(0,2*pi, 100);
plot3(cos(tvec), sin(tvec), 0*tvec + 3,'k-','linewidth',3)
plot3(2*cos(tvec), 2*sin(tvec), 0*tvec + 6,'k-','linewidth',3)
plot3(3*cos(tvec), 3*sin(tvec), 0*tvec + 11,'k-','linewidth',3)
title('Traces and Level Curves','fontsize',16)
```

MATLAB® Code: `Plotting_Paraboloid.m` in the Chapter 4 program repository.

```
%% This is Plotting_Paraboloid.m
% It plots the paraboloid z = x^2 + y^2 + 2 with the surf command.
clc; clf; clear; % clears console, figures, variables

%% The function z = f(x,y) x^2 + y^2 + 2
zfunction = @(x,y)[x.^2 + y.^2 + 2];

%% Plot the surface using the surf command
xvec = linspace(-3,3,40);  % create the vector of x-values
yvec = linspace(-3,3,40);   % create the vector of y-value
[x y] = meshgrid(xvec,yvec);  % create a full array of x & y values
z = zfunction(x,y);  % creates the full array of z-values
surf(x, y, z,'FaceColor', 'yellow');  % Plots the surface
title('Paraboloid - Surface','fontsize',16)
xlabel('x'); ylabel('y'); zlabel('z'); box on;
view(20,20); axis equal; axis([-4,4,-4,4,0,20]);
```

Chapter 4.1 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1.* Consider the surface defined by

$$z = \cos\left(\sqrt{x^2 + y^2}\right) \quad \text{for} \quad x \in [-2\pi, 2\pi] \quad \text{and} \quad y \in [-2\pi, 2\pi]$$

   (a) Sketch the trace of this surface in the xz-plane.

   (b) Sketch the trace of this curve in the yz-plane.

   (c) Sketch these traces in 3-Space.

   (d) Sketch the level curves for the following values of z.

      i. $z = 0$

      ii. $z = -1$

      iii. $z = 1$

   (e) Plot the surface using software

2. Consider the surface defined by

$$z = 4\, e^{-(x^2+y^2)} \quad \text{for} \quad x \in [-2, 2] \quad \text{and} \quad y \in [-2, 2]$$

   (a) Sketch the trace in the xz-plane.

   (b) Sketch the trace in the yz-plane.

   (c) Sketch these traces in 3-Space.

   (d) Describe the level curves for the following values of z.

      i. $z = 4$

      ii. $z = 2$

      iii. $z = 1$

   (e) Plot the surface using software

3.* Consider the surface defined by

$$z = f(x, y) = \frac{3}{1 + x^2 + y^2}$$

   (a) Define and sketch the traces in the xz and yz planes.

   (b) Use the traces to help plot the surface in 3-space.

4. Consider the surface defined by
$$z = f(x, y) = -4x^2 - y^2$$
   (a) Define and sketch the traces in the xz and yz planes.
   (b) Use the traces to help plot the surface in 3-space.

5.* Consider the plane defined implicitly by the equation $3x + 2y - 4z = 6$.
   (a) Express this plane explicitly in the form $z = f(x, y)$.
   (b) Express this plane in the form $F(x, y, z) = 0$.

6. Consider the surface defined implicitly by the equation $x^2 = y^2 + 7z^3$.
   (a) Express this surface explicitly in the form $z = f(x, y)$.
   (b) Express this surface in the form $F(x, y, z) = 0$.

7. Consider the sphere with radius 5 and centered at (2,1,-2) defined by the equation
$$(x-2)^2 + (y-1)^2 + (z+2)^2 = 25.$$
   (a) Express this sphere in the form $z = f(x, y)$.
   (b) Anything upsetting about your answer to (a)?
   (c) Express this sphere in the form $F(x, y, z) = 0$.

4.2 Partial Derivatives, Gradients, and Normal Vectors

When working with multi-variable functions, there is no longer a single derivative of the function. Instead, there are two or more **partial derivatives**. These can be used to find slopes of tangent lines to a surface in any direction instead of just the positive x-direction found in single-variable calculus. The partial derivatives also allow us to find gradients and normal vectors to surfaces which are nice to have when you start bouncing around in 3D. For now we will consider the only directions that are needed.

Partial Derivatives, Formal Definition

Given a multi-variable function of the form $z = f(x, y)$, the *partial derivative of f with respect to x*, denoted $\dfrac{\partial f}{\partial x}$ or f_x, is defined by

$$\frac{\partial f}{\partial x} = f_x(x, y) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \tag{4.6}$$

and likewise, the partial derivative of f with respect to y, denoted $\dfrac{\partial f}{\partial y}$ or f_y, is defined by

$$\frac{\partial f}{\partial y} = f_y(x, y) = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}. \tag{4.7}$$

Geometrically speaking, f_x is the slope (dz/dx) of the line parallel to the x-axis that is tangent to the surface $z = f(x, y)$. Likewise, f_y is the slope (dz/dy) of the line parallel to the y-axis that is tangent to the surface $z = f(x, y)$. These are both specific examples of *directional* derivatives. However, they turn out to be the only ones you really need.

Partial Derivatives, Informal Definition

When finding the partial derivative of a function with respect to a certain variable, you just differentiate with respect to that variable considering all other variables constant.

- **Example 1:** Consider the paraboloid defined by $z = f(x, y) = x^2 + y^2 + 2$.
  The partial derivatives are
  $$f_x(x, y) = 2x \quad \text{and} \quad f_y(x, y) = 2y$$
- **Example 2:** Consider the surface defined by $z = f(x, y) = 3x^2\, y^3$.
  The partial derivatives are
  $$f_x(x, y) = 6xy^3 \quad \text{and} \quad f_y(x, y) = 9x^2y^2$$
- **Example 3(harder):** Consider the surface defined by $z = f(x, y) = \cos\left(\sqrt{x^2 + y^2}\right)$.
  The partial derivatives are
  $$f_x(x, y) = \frac{-x \sin\left(\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}} \quad \text{and} \quad f_y(x, y) = \frac{-y \sin\left(\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}}$$

The Gradient Vector: The gradient of a function $f(x,y)$ is denoted ∇f.

$$\nabla f = \langle\ f_x,\ f_y\ \rangle \tag{4.8}$$

- **Example 1:** If $f(x,y) = x^2 + y^2 + 2$ then $\nabla f = \langle\ 2x,\ 2y\ \rangle$

- **Example 2:** If $f(x,y) = 3x^2\ y^3$. then $\nabla f = \left\langle\ 6xy^3,\ 9x^2y^2\ \right\rangle$

- **Example 3:** If $f(x,y) = \cos(\sqrt{x^2+y^2})$ then $\nabla f = \left\langle \dfrac{-x\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}},\ \dfrac{-y\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}} \right\rangle$

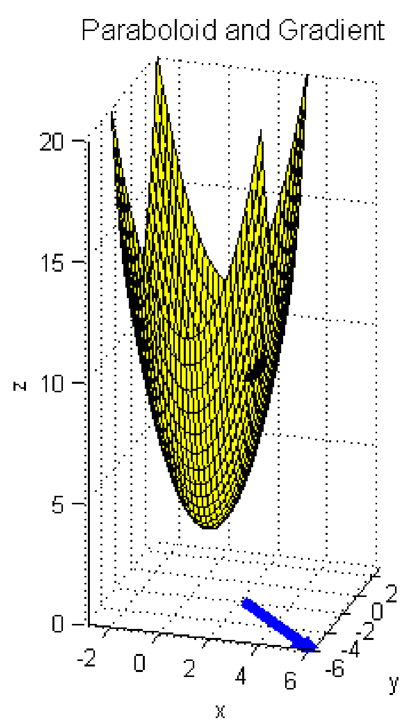
A practical implication: The gradient of a function $z = f(x,y)$ gives the direction (in the xy-plane) that leads to the greatest increase in z for a unit step-size.

- **Example:** If you are sitting on the surface $z = x^2 + y^2 + 2$ at the point (2,-2,10), what is the direction to move in the xy-plane that leads to the greatest increase?

  Answer: $\nabla f(2,-2) = \langle f_x(2,-2),\ f_y(2,-2)\ \rangle = \langle\ 4,\ -4\ \rangle$ depicted below.

MATLAB® code: `PlottingParaboloidAndGradient.m`

```
%% This is PlottingParaboloidAndGradient.m
clc; clf; clear;

%% The Surface Function
f = @(x,y)[x.^2 + y.^2 + 2];

%% Points and Gradient
P = [2,-2,10];     % The Point on the Surface
P2D = [2,-2,0];    % The Point in xy-plane
G = [4,-4,0];      % The Gradient in xy-plane

%% Plot The Surface
xvec = linspace(-3,3,30);  % range of x-values
yvec = linspace(-3,3,30);  % range of y-values
[x y] = meshgrid(xvec,yvec);  % array of x & y
z = f(x,y);  % create array of z-values
surf(x, y, z,'facecol', 'yellow');  hold on;

%% Plot the Point & Gradient
vectarrow(P2D,P2D+G,'blue');    % Plot gradient vector
plot3(P(1),P(2),P(3), 'k.','markersize',60) % point
xlabel('x'), ylabel('y'); zlabel('z'); % axis labels
view(17,22); axis equal;
title('Paraboloid and Gradient','fontsize',12);
```



The General Form of the Gradient and Normal Vectors

Suppose you have a surface defined by $F(x, y, z) = 0$, the gradient of F is

$$\nabla F = \langle\, F_x,\ F_y,\ F_z \rangle\,. \tag{4.9}$$

and $\pm\nabla F$ produces a vector that is normal to the surface defined by $F(x, y, z) = 0$.

Practical Usage: If you have a surface defined by $z = f(x, y)$, this surface can be defined by $f(x, y) - z = 0$ or $z - f(x, y) = 0$. Using these expressions for $F(x, y, z)$, the normal vectors to the surface are

$$\mathbf{n}_1 = \langle f_x,\ f_y,\ -1\rangle \quad \text{and} \quad \mathbf{n}_2 = \langle -f_x,\ -f_y,\ 1\rangle\,. \tag{4.10}$$

- **Example:** Consider the surface defined by $z = x^2 + y^2 + 2$.
  Find the normal vector to the surface at the point $(2, -2, 10)$ that is pointing down.

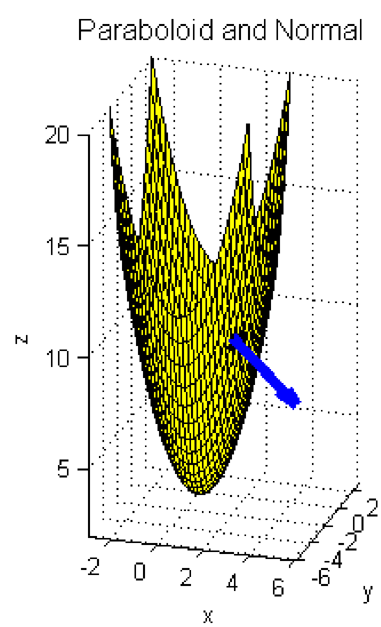
  Answer: We choose the $\mathbf{n}_1$ normal from equations (4.10) because it has the negative z component.

  $$\mathbf{n}_1 = \langle f_x,\ f_y,\ -1\rangle = \langle\ 2x,\ 2y,\ -1\ \rangle$$

  Evaluating this at the point (2,-2,10) gives $\mathbf{n}_1 = \langle 4,\ -4,\ -1\ \rangle$. The surface and this normal are depicted in the graph from the following code.

MATLAB® code: `PlottingParaboloidAndNormal.m`

```
%% This is PlottingParaboloidAndNormal.m
clc; clf; clear;

%% The Surface Function
f = @(x,y)[x.^2 + y.^2 + 2];

%% The Point and Normal
P = [2,-2,10];     % The Point on the Surface
n = [4,-4,-1];     % The normal vector

%% Plot The Surface
xvec = linspace(-3,3,30);  % range of x-values
yvec = linspace(-3,3,30);  % range of y-values
[x y] = meshgrid(xvec,yvec);  % array of x & y
z = f(x,y);  % create array of z-values
surf(x, y, z,'facecol', 'yellow');  hold on;

%% Plot the Point & Normal Vector
vectarrow(P,P+n,'blue');    % Plot normal vector
plot3(P(1),P(2),P(3), 'k.','markersize',40) % point
xlabel('x'), ylabel('y'); zlabel('z'); % axis labels
view(17,22); axis equal;
title('Paraboloid and Normal','fontsize',12)
```



Chapter 4.2 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

1. Find the partial derivatives f_x and f_y for the following functions.

   (a)* $f(x, y) = xy + x^2 + y^2$

   (b) $f(x, y) = \cos(x^2 + y^2)$

   (c)* $f(x, y) = xy\cos(2x)$

   (d) $f(x, y) = ye^{2xy}$

   Find the partial derivatives F_x, F_y, and F_z for the following functions.

   (e)* $F(x, y, z) = 3xy^2z^3$

   (f) $F(x, y, z) = e^{xy}\sin(2z)$

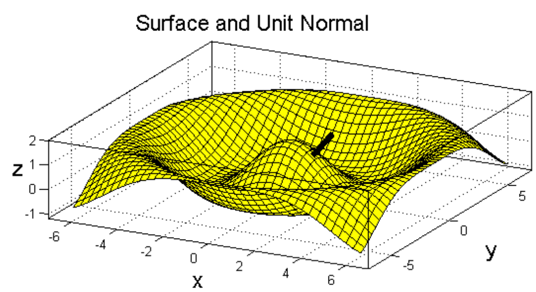
2.* Consider the surface defined by

$$z = f(x, y) = \cos\left(\sqrt{x^2 + y^2}\right) \quad \text{for} \quad x \in [-2\pi, 2\pi] \quad \text{and} \quad y \in [-2\pi, 2\pi]$$

   This is the same function used in Example 3 at the beginning of this section.

   (a) Find f_x and f_y.

   (b) Find the **unit** normal vector to the surface at the point $(1, 0, f(1, 0))$ depicted below.



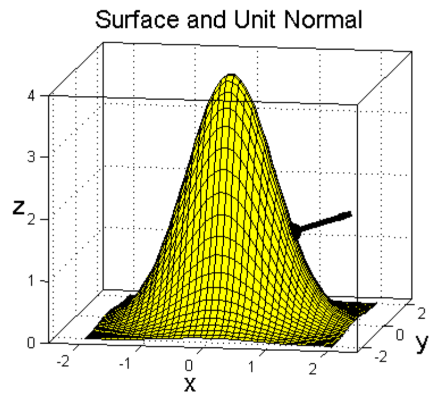
   (c) Use MATLAB® to make the graph above. You can start with `PlottingParaboloidAndNormal.m` in the Chapter 4 program repository. Don't forget to normalize the normal vector (give it length one) and use `axis equal;` to make the normal look normal.

3. Consider the surface defined by

$$z = f(x,y) = 4\, e^{-(x^2+y^2)} \quad \text{for} \quad x \in [-2,2] \quad \text{and} \quad y \in [-2,2]$$

(a) Find f_x and f_y.

(b) Find the **unit** normal vector to the surface at the point $(1, 0, f(1,0))$ depicted below.



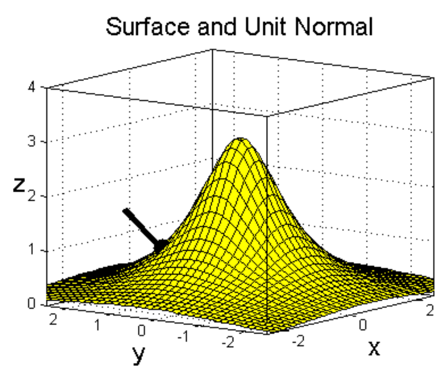
(c) Use MATLAB® to make the graph above. You can start with `PlottingParaboloidAndNormal.m` in the Chapter 4 program repository. Don't forget to normalize the normal vector (give it length one) and use `axis equal;` to make the normal look normal.

4.* Consider the surface defined by

$$z = f(x,y) = \frac{3}{1+x^2+y^2} \quad = \quad 3\left(1+x^2+y^2\right)^{-1}$$

(a) Find f_x and f_y.

(b) Find the **unit** normal vector to the surface at the point $(-1, 1, 1)$ depicted below.



5. Consider the surface defined by
$$z = f(x,y) = -4x^2 - y^2$$
   (a) Find f_x and f_y.
   (b) Find the **unit** normal vector to the surface at the point $(1,1,-5)$ that points in the positive z-direction.

6. Find the gradient of the given function.
   (a)* $F(x,y,z) = 3x^2 - 2y - 3\sin(z) - 12$
   (b) $F(x,y,z) = 3x^2yz^2$

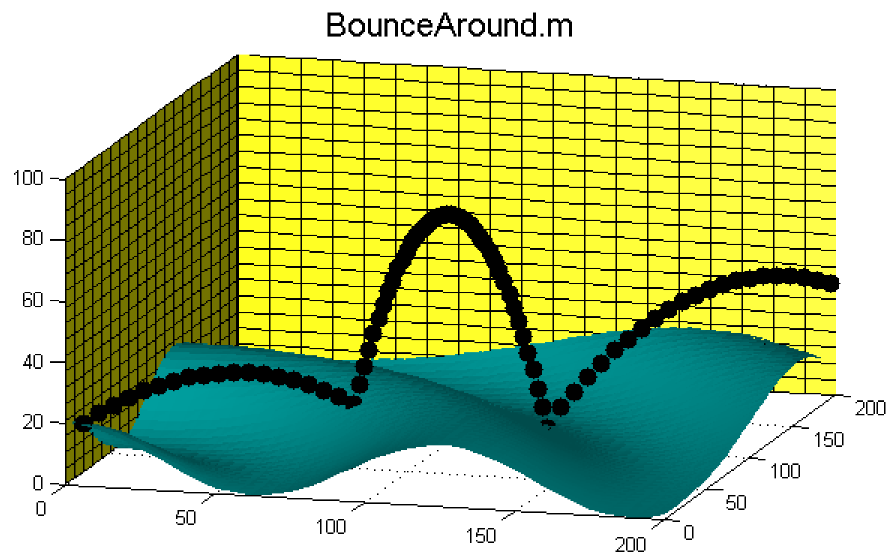
7. Consider the plane defined by
$$\text{plane:} \quad 3x + 2y - 4z = 6$$
   Find two **unit** normal vectors at the point (2,0,0) via two different methods.
   (a) Express the plane as $z = f(x,y)$ and then $\mathbf{n} = \pm\langle f_x, f_y, -1\rangle$.
   (b) Express the plane as $F(x,y,z) = 0$ and then $\mathbf{n} = \pm\langle F_x, F_y, F_z\rangle$.

8.* Consider the sphere with radius 5 and centered at (2,1,-2) defined by the equation
$$(x-2)^2 + (y-1)^2 + (z+2)^2 = 25.$$
   Find a normal vector to the surface of the sphere at the point (2,4,-6) which points away from the center via the following two different methods.
   (a) Express the part of the sphere with point (2,4,-6) as $z = f(x,y)$, find $\mathbf{n} = \pm\langle f_x, f_y, -1\rangle$, and determine which one ($\pm$) points away from the center.
   (b) Express the sphere as $F(x,y,z) = 0$ and then $\mathbf{n} = \pm\langle F_x, F_y, F_z\rangle$.

9. Consider the surface defined implicitly by the equation $x^2 = y^2 + 7z^3$. Find a normal vector to the surface at the point (4,3,1).

4.3 Bouncing Around in 3D

In this chapter we will bounce objects around in 3-space. We will bounce them off of surfaces that are not necessarily planes. The bounce vectors will be based on the incoming trajectory with respect to the normal vector at the point of contact. With this ability and Euler's Method for moving around in real-time, we can then bounce around in space until something happens .. or not.



Important Stuff

- You move around via Euler's method. (let $\mathbf{r}_n$ be the current location, $\mathbf{v}_n$ be the current velocity vector, Δt be the time step, and $\mathbf{r}_{n+1}$ be the next position):

  $$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta t \, \mathbf{v}_n$$

  If acceleration is constant (gravity), there is no air resistance, and no external forces, then

  $$\mathbf{v}_n = \mathbf{v}_{n-1} + \Delta t \, \langle 0, 0, -g \rangle \quad \text{and}$$

  where g is the acceleration due to gravity (32 feet/sec^2, or 9.8 m/sec^2).

- If you hit something with unit normal vector $\mathbf{n}$, and damping term D:

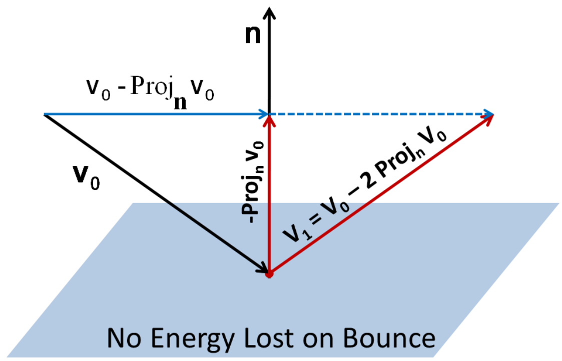
  $$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta t \, (\mathbf{v}_n - 2\text{Proj}_{\mathbf{n}}\mathbf{v}_n) \, D$$

- Be very careful about leaving the playing area. You might go away and never come back. This can happen if the velocity gets too great or the time step is too large.

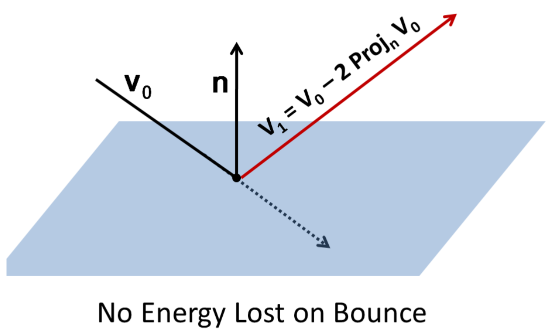
A Bounce in 3D

Suppose an object hits a surface with an incoming velocity vector of $\mathbf{v}_0$ at a point with a normal vector $\mathbf{n}$ (found using the gradient of the function) and we want it to bounce off of the surface in a appropriate manner. The bounce vector, denoted $\mathbf{v}_1$ is found using the formula

$$\mathbf{v}_1 = \mathbf{v}_0 - 2\ \text{Proj}_{\mathbf{n}}\mathbf{v}_0 \tag{4.11}$$



If $\mathbf{v}_0$ hits the surface perfectly
$\mathbf{v}_1 = \mathbf{v}_0 - 2\ \text{Proj}_{\mathbf{n}}\mathbf{v}_0$



Even if
$\mathbf{v}_0$ goes through the surface
$\mathbf{v}_1 = \mathbf{v}_0 - 2\ \text{Proj}_{\mathbf{n}}\mathbf{v}_0$

Finding the actual point of contact with the surface is tricky!

Equation (4.11) remains valid regardless of your choice for $\mathbf{n}$ so long as it is indeed normal to the surface.

Damping: If there is energy lost on the bounce, then the bounce vector will have a smaller magnitude than the incoming vector. This can be modeled with a damping coefficient applied to equation (4.11) as

$$\mathbf{v}_1 = D\ \left(\mathbf{v}_0 - 2\ \text{Proj}_{\mathbf{n}}\mathbf{v}_0\right), \tag{4.12}$$

for $D \in (0, 1]$. If $D = 1$, there is no damping and no energy lost on the bounce. Otherwise,

$$||\mathbf{v}_1|| = D\ ||\mathbf{v}_0||.$$

MATLAB® code: This file, `BounceAround.m`, creates the bouncing ball animation depicted on the first page of this section. It calls `MySurface.m`, `MySurfaceGradient.m`, and `collisionDSurface.m` function files. These last three files are displayed on the next page. All files are in the Chapter 4 program repository.

```
%% This is BounceAround.m
% Performs a bouncing ball simulation with crude real-time collision detection.
clf; clc; clear; % clears command window, figure, and variables
%% Functions I'm calling from other files
%PlotPlaneFunction.m, ProjUV.m, collisionDSurface.m, MySurface.m, MySurfaceGradient.m

%% Initial Values
D = 1;   % this is the damping on a bounce D in (0,1]
tend = 8; tsteps = tend*10;  Delta_T = tend/(tsteps-1);  %total time and time steps
vo = 60;  % initial speed (in feet per second).
thetadegrees = 45; phidegrees = 25; % angle from positive x axis and elevation angle.
theta=thetadegrees*pi/180; phi = phidegrees*pi/180;   % convert to radians
xyzbounds=[0,200,0, 200, 0, 100];  %plot bounds for PlaneFunction, collisionDSurface
Ro = [5,5,MySurface(5,5)];     % Initial Point on the Surface.
Vo = vo*[cos(theta)*cos(phi), sin(theta)*cos(phi), sin(phi)];  % Initial Velocity

%% Create the surface using the surf command and the planes that make the walls
xvec = linspace(xyzbounds(1),xyzbounds(2),40);
yvec = linspace(xyzbounds(3),xyzbounds(4),40);
[xsurf, ysurf] = meshgrid(xvec,yvec);  % mesh out x and y
zsurf = MySurface(xsurf,ysurf);    % creates z-values for the bottom surface.
surf(xsurf,ysurf,zsurf,'FaceColor','cyan','EdgeColor','none'); %plots surface.
hold on; camlight right; %provides good lighting for curved surface
PlotPlaneFunction([1,0,0],[0,0,0],xyzbounds,'yellow'); % plots a wall
PlotPlaneFunction([0,-1,0],[0,200,0],xyzbounds,'yellow'); %plots another wall
pointplot = plot3(Ro(1),Ro(2),Ro(3),'k.','markersize',30); %plots the point.
axis(xyzbounds);  view(16,12); axis equal;
title('click to start','fontsize',16); waitforbuttonpress;

%%  Animation Loop
t = 0;
for i = 1:1:tsteps-1
  t = t + Delta_T;
  R1 = Ro + Delta_T*Vo;
  text = sprintf('Running: t = %1.2f, R1 = (%1.1f, %1.1f, %1.1f)',...
      t, R1(1), R1(2), R1(3));
  title(text,'fontsize',16);
  [R1,V1] = collisionDSurface(R1,Vo,Delta_T,xyzbounds,D);  % collision detection.
  delete(pointplot)  % deletes previous ball
  pointplot = plot3(R1(1),R1(2),R1(3),'k.', 'markersize',30);
  Ro = R1; Vo = V1;   %/ the new position and velocity vectors.
  pause(.01);
end
text = sprintf('Done: Delta t = %1.2f, tend = %1.2f, R1 = (%1.1f, %1.1f, %1.1f)', ...
    Delta_T,t,R1(1), R1(2), R1(3));
title(text,'fontsize',16);  disp('done')
```

MATLAB® function file: `MySurface.m`

This defines the bottom surface. It's used in `BounceAround.m` and `collisionDSurface.m`.

```
function z = MySurface(x,y)
% Defines the Surface in BounceAround.m
  z = 10*cos(0.05*sqrt(x.^2 + y.^2))+10;
```

MATLAB® function file: `MySurfaceGradient.m`

This uses the gradient to calculate the unit normal vector to the surface. It's used in `collisionDSurface.m`.

```
function u = MySurfaceGradient(x,y,z)
% Gets the unit normal to my surface in bounce around
  fx = -10*(.05)*x*sin(.05*sqrt(x^2 + y^2))/sqrt(x^2+y^2);
  fy = -10*(.05)*y*sin(.05*sqrt(x^2 + y^2))/sqrt(x^2+y^2);
  n_down = [fx,fy,-1];      % downward pointing normal
  n_up = [-fx,-fy,1];       % upward pointing normal
  n = n_up; u = n/norm(n);  % choose up and normalize it
```

MATLAB® function file: `collisionDSurface.m`

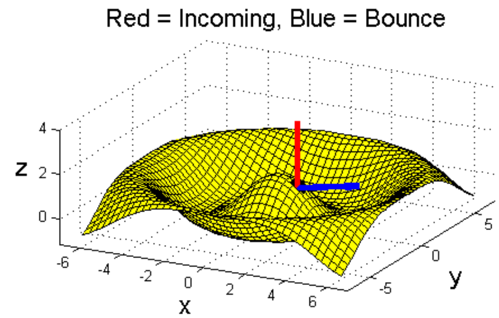
This determines whether or not there has been a collision. If so, it determines the normal vector, places the ball on the surface where it hit, and calculates the bounce velocity vector.

```
function [R1,V1] = collisionDSurface(R1,Vo,Delta_T,xyzbounds,D);
% The collision detection function which finds the new velocity vector.
  x = R1(1); y=R1(2); z=R1(3);
  xmin = xyzbounds(1); ymin = xyzbounds(3); zmin = MySurface(x,y);
  xmax = xyzbounds(2); ymax = xyzbounds(4); zmax = xyzbounds(6);
  if x > xmin & x < xmax & y > ymin & y < ymax & z > zmin & z < zmax
    V1 = Vo - [0,0,32*Delta_T];  % Euler's method on velocity, no bounce
    R1 = R1;   % No change to R1
  else  %there has been a collision with one of the surfaces
    if x <= xmin
        n = [1,0,0]; x = xmin;  % normal and point on left wall
    elseif x>= xmax
        n = [-1,0,0]; x = xmax; % normal and point on right wall
    elseif y <= ymin
        n = [0,1,0]; y = 0;     % normal and point on near wall
    elseif y>= ymax
        n = [0,-1,0]; y = ymax; % normal and point on far wall
    elseif z >= zmax
        n = [0,0,-1]; z = zmax; % normal and point on the ceiling
    else
        n = MySurfaceGradient(x,y,z); z = zmin; % normal and point on surface
    end
    V1 = D*(Vo - 2*ProjUV(Vo,n)); % new velocity vector after bounce
    R1 = [x,y,z];  % new R1 on surface where it hit.
  end
```

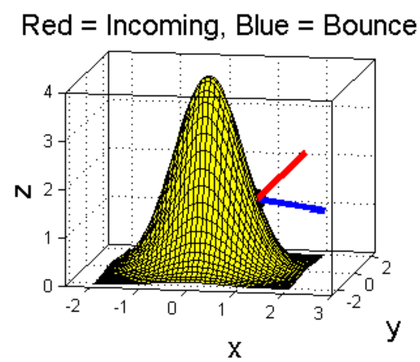
Chapter 4.3 Problem Set

Numbers with an asterisk* have solutions in the back of the book.

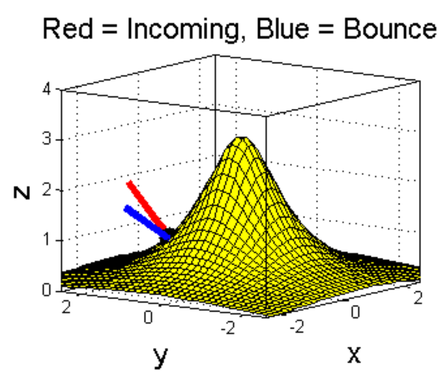
1.* Consider the surface defined by
$z = f(x, y) = \cos\left(\sqrt{x^2 + y^2}\right)$.
Suppose a ball hits this surface at the point $(1, 0, \cos(1))$ with an incoming velocity vector of $\mathbf{v}_o = \langle 0, 0, -3 \rangle$. Find the bounce vector $\mathbf{v}_1$ with no damping ($D = 1$). The graph to the right depicts this bounce.



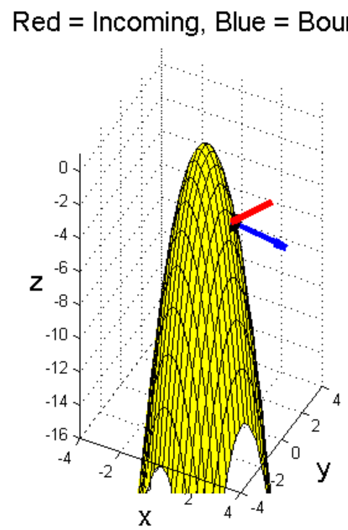
2. Consider the surface defined by
$z = f(x, y) = 4\, e^{-(x^2+y^2)}$.
Suppose a ball hits this surface at the point $(1, 0, f(1, 0))$ with an incoming velocity vector of $\mathbf{v}_o = \langle -1, 0, -1 \rangle$. Find the bounce vector $\mathbf{v}_1$ with no damping ($D = 1$). The graph to the right depicts this bounce.



3.* Consider the surface defined by
$z = f(x, y) = \dfrac{3}{1 + x^2 + y^2}$.
Suppose a ball hits this surface at the point $(-1, 1, 1)$ with an incoming velocity vector of $\mathbf{v}_o = \langle 0, -1, -1 \rangle$. Find the bounce vector $\mathbf{v}_1$ with no damping ($D = 1$). The graph to the right depicts this bounce.



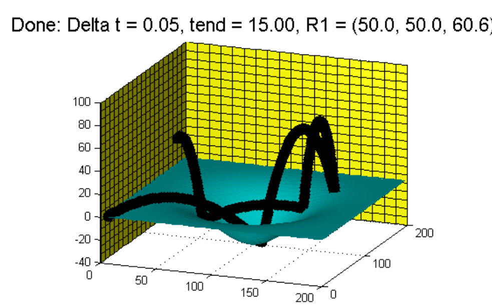
4. Consider the surface defined by
$z = f(x, y) = -4x^2 - y^2$.
Suppose a ball hits this surface at the point $(1, 1, -5)$ with an incoming velocity vector of $\mathbf{v}_o = \langle -2, 0, -2 \rangle$. Find the bounce vector $\mathbf{v}_1$ with no damping ($D = 1$). The graph to the right depicts this bounce.



5.* Suppose a ball hits the plane defined by $2x + 2y - z = 12$ at the point (4,3,2) with an incoming velocity of $\mathbf{v}_0 = \langle 2, 3, 6 \rangle$. Find the bounce vector $\mathbf{v}_1$ with no damping ($D = 1$).

6. Suppose a ball hits the plane defined by $x - 2y + 3z = 2$ at the point (1,1,1) with an incoming velocity of $\mathbf{v}_0 = \langle 1, -2, 2 \rangle$. Find the bounce vector $\mathbf{v}_1$ if 20% of the energy is lost after the bounce ($D = 0.80$).

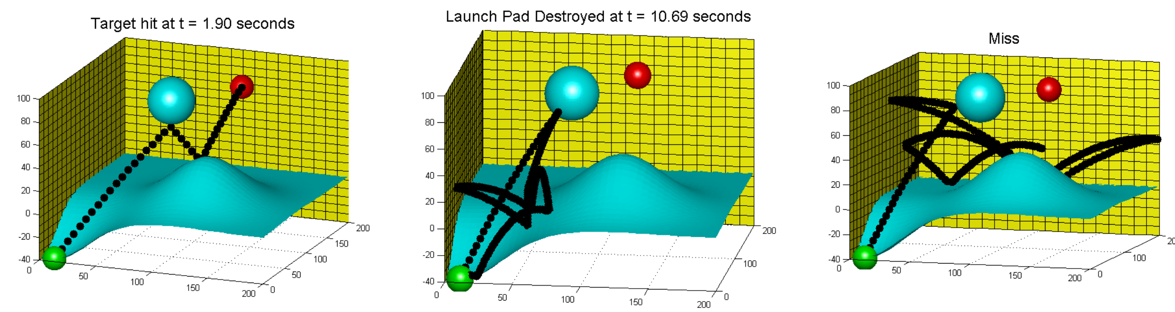
7. (MATLAB®) Create a bouncing ball simulation similar to the one presented in this chapter. Change the surface from
$f(x, y) = 10\ \cos\left(0.05\ \sqrt{x^2 + y^2}\right) + 10$
to
$f(x, y) = k\ e^{r\ \left[(x-100)^2+(y-100)^2\right]}$,
where $k = -40$ and $r = -0.0005$.



You'll need minimal editing to `BounceAround.m` but you will have to change the functions in `MySurface.m` and the derivatives in `MySurfaceGradient.m`.

Chapter 4 Project: Projectile Game in 3D

Create some type of 3D game with a projectile and a target. The projectile must follow a trajectory described by Newtonian Physics as estimated by Euler's Method. There should be some type of bouncing and some type of collision detection to determine whether or not a target has been hit. This type of game often takes the form of a tank shooting at a target. You are free to change this theme to one of your choosing. Be creative. Below are screenshots from an example.



The spheres in this example are created by using the function file **`build_sphere.m`** available from the Chapter 4 program repository. This function file builds the array of values for a sphere with a given center and radius. You can use the returned variables in the **`surf`** plotting command.

<u>MATLAB® function file:</u> `build_sphere.m`

```
function [x,y,z] = build_sphere(center, radius)
% Input the center [xo,yo,zo] and radius of the desired sphere.
% Returns x, y, and z to plot with the surf or mesh function.

[xl yl zl] = sphere(100);    % Built-in sphere function.
                             % Creates unit sphere at (0,0,0)
x = radius*xl + center(1);
y = radius*yl + center(2);   % Expand radius and re-center.
z = radius*zl + center(3);
% You can now plot the sphere with surf(x,y,z).
% For example, to plot a sphere with center [3,4,5] with radius 10,
% [x,y,z] = build_sphere([3, 4, 5], 10)
% surf(x,y,z)
```

Appendix A

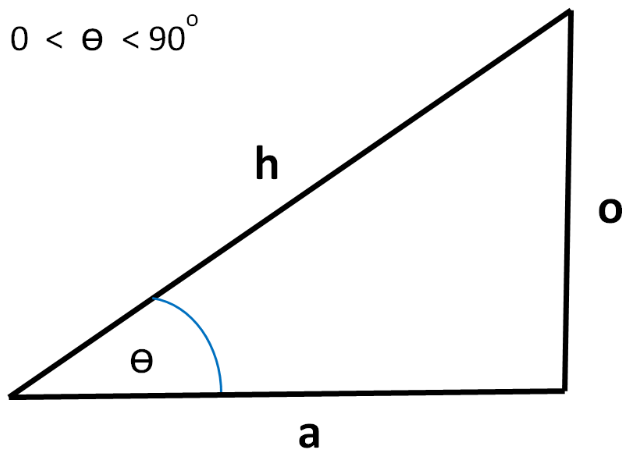
Trigonometry Review

In this appendix, we start briefly with some *triangle trigonometry*, and then move onto *unit-circle trigonometry*, and then to trigonometry as periodic functions of a continuous variable. We end with how to create circles and ellipses and a brief description of the tangent function.

Contents

A.1 Triangle Trigonometry

Consider the right triangle below. We focus here on three trigonometric functions: cosine (cos), sine (sin), and tangent (tan). These functions are defined in terms of an angle θ (theta).



$$\sin(\theta) = \frac{o}{h}$$

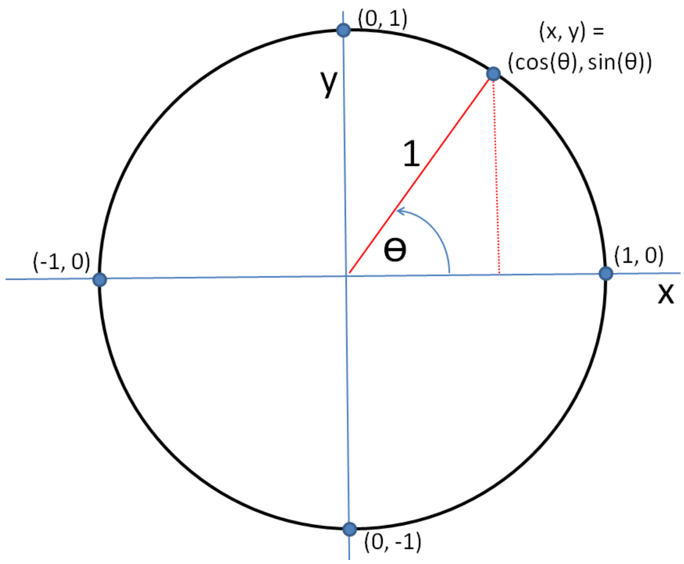
$$\cos(\theta) = \frac{a}{h}$$

$$\tan(\theta) = \frac{o}{a}$$

- There are quite a few things you can do with this.
  - If you know two of the side lengths, you can utilize Pythagorean's theorem ($a^2 + o^2 = h^2$) to get the third side length and hence all of the trig functions of all of the angles. Through inverse trig functions, you can get both of the unknown angles as well.
  - If you know θ and one side length, you can, through various identities and inverse trig functions determine the other two side lengths, all of the trig functions of that angle, and the other angle.
  - Once you include the law of sines and/or the law of cosines you can start to play with non-right triangles as well.
- However, there is a lot you can not do with these relations.
  - Notice, with this limited view of the trigonometric functions, the angle (θ) must be between 0 and 90^o.
  - Non-right triangles are very difficult to handle. If, by chance, you recall the law of cosines, you'll remember that it was not a pleasant formula.
  - You would not immediately recognize that these trigonometric functions are periodic and that is very important.

A.2 Unit-Circle Trigonometry

Consider the circle centered at (0, 0) in the cartesian plane with radius equal to one. Now we define our trig functions in terms of the angle traced out by the ray moving counter-clockwise around the circle.



For every point (x, y) on the unit circle:

$$\cos(\theta) = x,$$

$$\sin(\theta) = y,$$

$$\tan(\theta) = \frac{y}{x}.$$

- Notice: These match the triangle trig functions when $0 < \theta < 90^o$ (because $h = 1$).
- We can use any angle we want, even negative angles.
- It is immediately obvious what $cos(\theta)$ and $\sin(\theta)$ are for $\theta = 0, 90, 180, 270, 360, \ldots$.
- It is obvious that sine and cosine functions repeat themselves after every full rotation. In other words, these functions are periodic.
- Radians: Instead of measuring θ in degrees we now measure it with respect to the arc-length traced out by the unit-circle to the point $(x,\ y)$. This type of angle measurement is called radians. One full revolution is $360^o = 2\pi$ radians. A half a revolution is $180^o = \pi$ radians. A quarter revolution is $90^o = \pi/2$ radians. Almost all calculators and software calculate trig functions assuming the argument is in radians.
- **Converting between Degrees and Radians**: If r is radians and d is degrees, then

$$d = \frac{180}{\pi}\, r \quad \text{and} \quad r = \frac{\pi}{180}\, d.$$

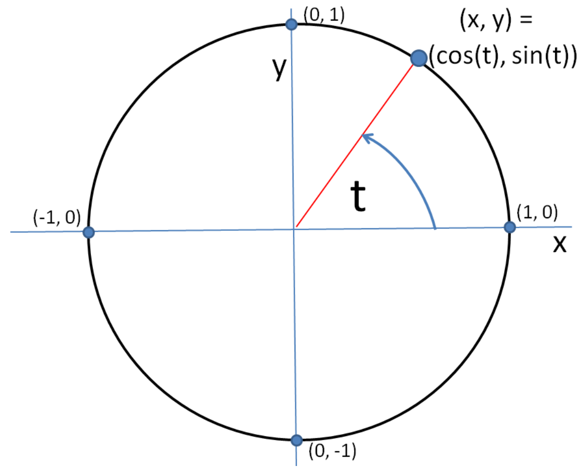
- If the radius is r instead of one then triangle trigonometry gives us

$$\cos(\theta) = \frac{x}{r} \qquad \text{and} \qquad x = r\cos(\theta), \tag{A.1}$$

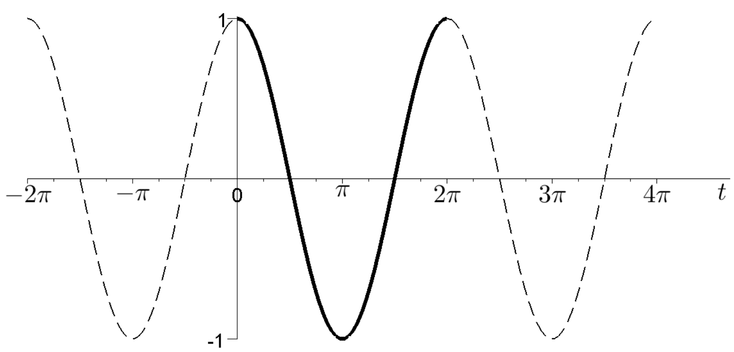
$$\sin(\theta) = \frac{y}{r} \qquad \text{and} \qquad y = r\sin(\theta). \tag{A.2}$$

A.3 Trigonometry as a Collection of Periodic Functions

Here we look at cosine and sine as periodic functions of a continuous variable (we will save tangent for later). Consider the unit circle as the angle (now denoted by t in **radians**) goes around the circle in the counterclockwise direction. If we track $x = \cos(t)$ and $y = \sin(t)$ to plot these functions, we get the following periodic graphs.



$\cos(t)$

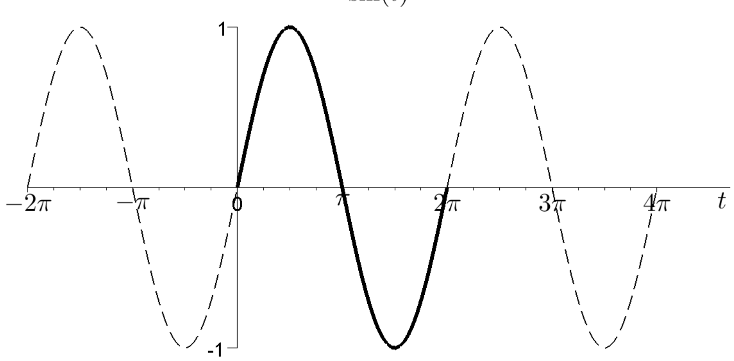


$\cos(t)$ has period 2π.
For $k =$ any integer,

$$\cos(t + k\ 2\pi) = \cos(t),$$

$$\cos(2k\ \pi) = 1,$$
$$\cos(\ (2k+1)\ \pi) = -1,$$
$$\cos\left(\ \left(\tfrac{2k+1}{2}\right)\ \pi\right) = 0.$$

$\sin(t)$



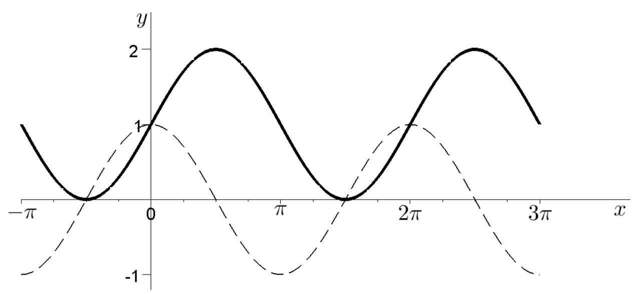
$\sin(t)$ has period 2π.
For $k =$ any integer,

$$\sin(t + k\ 2\pi) = \sin(t),$$

$$\sin(k\ \pi) = 0,$$
$$\sin\left(\ \left(\tfrac{4k+1}{2}\right)\ \pi\right) = 1.$$
$$\sin\left(\ \left(\tfrac{4k-1}{2}\right)\ \pi\right) = -1.$$

A.4 Translations and Transformations of Trig Functions

Horizontal and Vertical Translations: $y = \cos(x - \phi) + B$ and $y = \sin(x - \phi) + B$



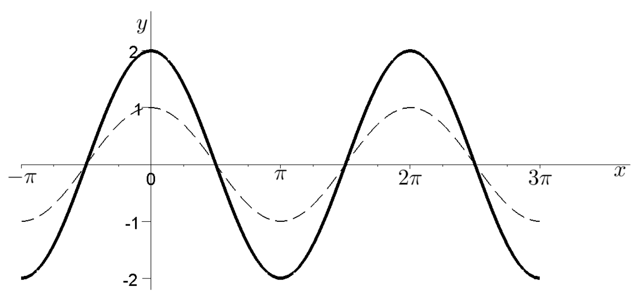
Dashed: $y = \cos(x)$.

Solid: $y = \cos(x - \pi/2) + 1$

Horizontal shift by $\pi/2$

Vertical shift by 1.

Amplitude Changes: $y = A\ \cos(x)$ and $y = A\ \sin(x)$, where $A =$ Amplitude.



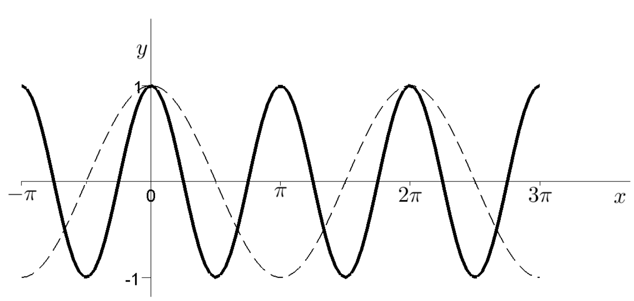
Dashed: $y = \cos(x)$.

Solid: $y = \mathbf{2}\cos(x)$

Vertical Stretch by 2.

Amplitude increases.

Period (Frequency) Changes: $y = \cos(\omega x)$ and $y = \sin(\omega x)$, where the period $= \dfrac{2\pi}{\omega}$.



Dashed: $y = \cos(x)$.

Solid: $y = \cos(\mathbf{2}x)$

Period $= (2\pi/\mathbf{2}) = \pi$

The period decreases
and the frequency increases.

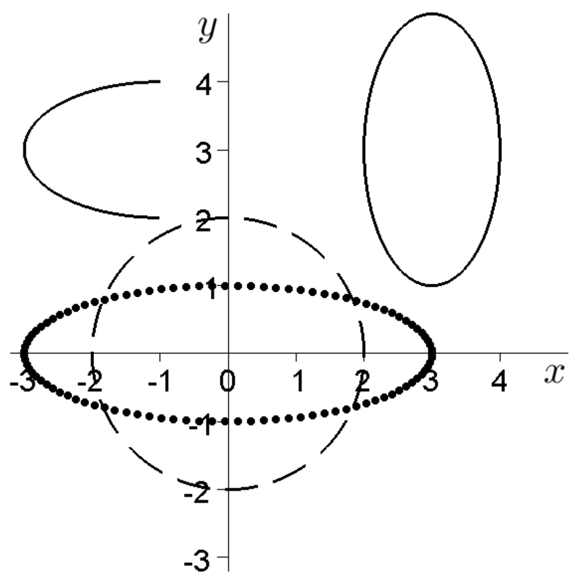
A.5 Circles and Ellipses

Getting the graph of a circle in terms of $y = f(x)$ is tricky and defining the graph of an ellipse is even trickier. These curves are much easier to create when you define them as a set of trigonometric **parametric** equations. In a parametric curve, the values of x and y are both determined in terms of another variable (parameter) usually denoted as t or θ.

An ellipse with center (x_o, y_o), x-radius of r_x, and y-radius of r_y is defined by

$$x(t) = x_o + r_x \cos(t), \quad y(t) = y_o + r_y \sin(t), \quad t \in [0, 2\pi] \tag{A.3}$$

To make partial ellipses and circles, let the parameter range over an appropriate subset.

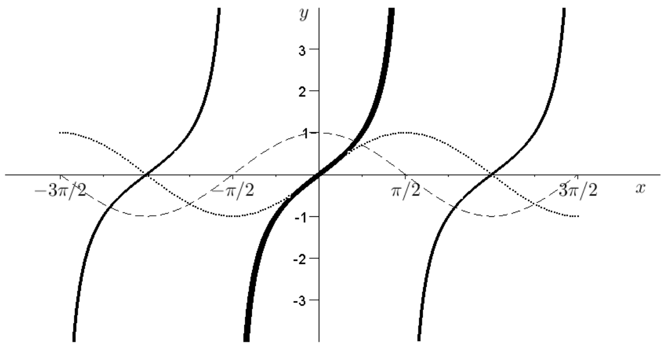


Dashed (circle)	**Dotted (ellipse)**	**Solid (ellipse)**	**Solid (half-ellipse)**
$x(t) = 2\cos(t)$	$x(t) = 3\cos(t)$	$x(t) = 3 + \cos(t)$	$x(t) = -1 + 2\cos(t)$
$y(t) = 2\sin(t)$	$y(t) = \sin(t)$	$y(t) = 3 + 2\sin(t)$	$y(t) = 3 + \sin(t)$
$t \in [0, 2\pi]$	$t \in [0, 2\pi]$	$t \in [0, 2\pi]$	$t \in [\pi/2, 3\pi/2]$

A.6 The Tangent Function

The tangent function is defined in terms of the cosine and sine functions by

$$\tan(x) = \frac{\sin(x)}{\cos(x)}. \tag{A.4}$$



Dashed: $\cos(x)$

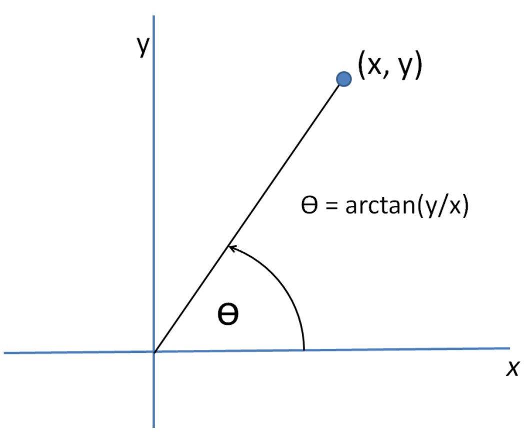
Dotted: $\sin(x)$

Solid: $\tan(x)$

$\tan(x) = \frac{\sin(x)}{\cos(x)}$

$\tan = \frac{\text{dotted}}{\text{dashed}}$

- Two properties of the tangent function can be seen in its graph.
  - Unlike sine and cosine, tangent has a period of π rather than 2π.
  - The tangent function is undefined at $\frac{2k+1}{2}\pi$ for all integers k, and the graph of $\tan(x)$ has vertical asymptotes at these locations.
- The **arctangent** function is the inverse of the tangent function (sometimes denoted $\tan^{-1}$).



Very useful, but

$-\pi/2 < \arctan \leq \pi/2$.

Get problems if $x \leq 0$.

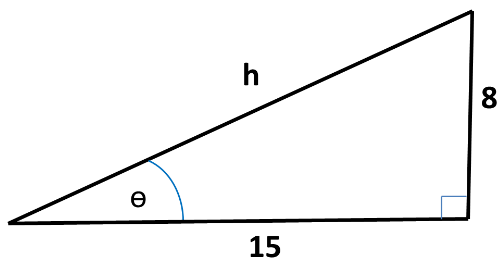
Better to use `atan2(y,x)` when available*.

$-\pi <$ `atan2(y,x)` $\leq \pi$

* Most calculators and software contain an `atan2` function, which resolves all of the issues of using the arctan function when $x \leq 0$.

A.7 Trigonometry Review - Problem Set

1. Find the requested information for the right triangle below.



$h =$

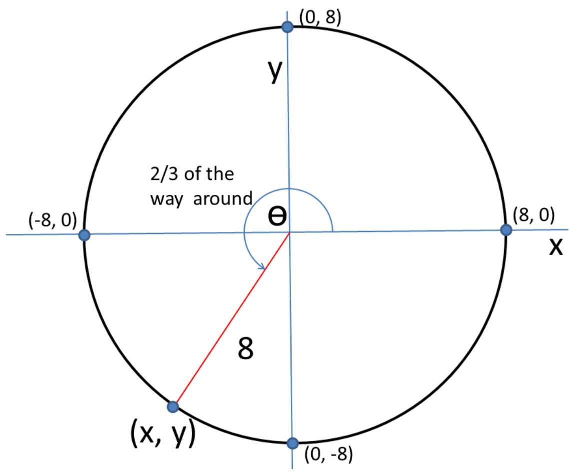
$\sin(\theta) =$

$\cos(\theta) =$

$\tan(\theta) =$

$\theta =$

2. Suppose θ takes you 2/3 of the way around the circle centered at (0,0) with radius 8. Find the angle θ (in degrees and radians) and the point (x, y) on the circle.



θ (degrees) $=$

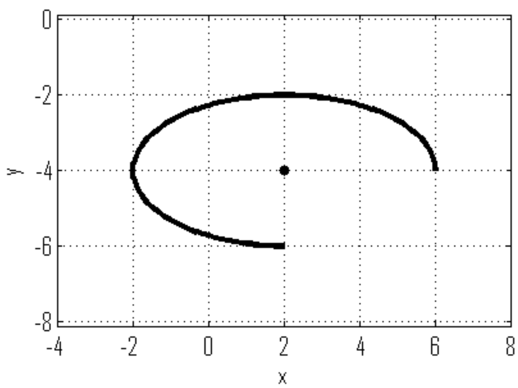
θ (radians) $=$

$x =$

$y =$

3. Consider $y = A\cos(\omega t) + B$. Find A, ω, and B that produces an oscillating periodic function with period $= 3$, that oscillates between 4 and 8 on the y-axis. Plot this function for $t \in [-3, 6]$.

4. Find the parametric equations for the ellipse centered at (2,-4) pictured below. Make sure the curve is traced in the counter-clockwise direction as the parameter t increases.



$x(t) =$

$y(t) =$

$t \in$

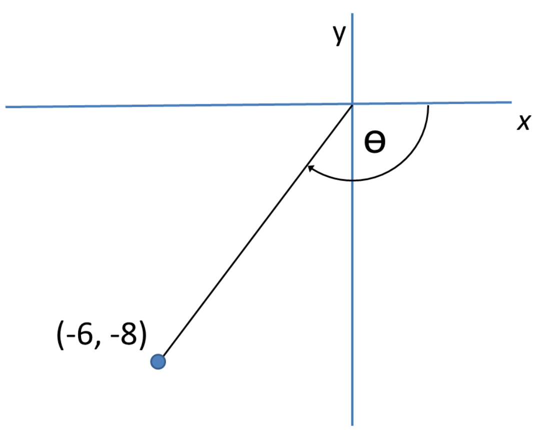
5. Sketch the graph of the ellipse defined by the parametric equations:

$$x = 4 + 5\cos(t)$$

$$y = -2 + 2\sin(t)$$

$$t \in [0, 2\pi]$$

6. Find θ in radians and degrees. This will be a negative angle.



θ (radians) $=$

θ (degrees) $=$

Appendix B

Review of Differentiation Rules

This appendix contains a list of the basic differentiation rules you should have seen in Calculus I.

Notation: If u is a differentiable function of x then the derivative of u with respect to x is denoted by

$$\frac{du}{dx} = \frac{d}{dx}u = u'$$

General Single-Variable Differentiation Rules:
Assume u and v are differentiable functions of x, and c is a constant.

- $\dfrac{d}{dx}c = 0$
- $\dfrac{d}{dx}(cu) = cu'$
- $\dfrac{d}{dx}(u+v) = u' + v'$
- Chain Rule: $\dfrac{d}{dx}u(v(x)) = \dfrac{du}{dv}\,\dfrac{dv}{dx} = \dfrac{du}{dv}\,v'$
- Product Rule: $\dfrac{d}{dx}[\,u\,v\,] = u\,v' + v\,u'$
- Quotient Rule: $\dfrac{d}{dx}\left[\dfrac{u}{v}\right] = \dfrac{vu' - uv'}{v^2}$

Polynomials and the Power Rule

Assume u is a differentiable functions of x and n is any real number not equal to zero.

- $\dfrac{d}{dx}x^n = n\, x^{n-1}$ provided $n \neq 0$.
- $\dfrac{d}{dx}\left[\, u^n \,\right] = n\, u^{n-1}\, u'$ provided $n \neq 0$

Trigonometric Functions

Assume u is a differentiable function of x.

- $\dfrac{d}{dx}\left[\, \sin u \,\right] = (\cos u)\, u'$
- $\dfrac{d}{dx}\left[\, \cos u \,\right] = -(\sin u)\, u'$
- $\dfrac{d}{dx}\left[\, \tan u \,\right] = (\sec^2 u)\, u'$
- $\dfrac{d}{dx}\left[\, \sec u \,\right] = (\sec u\ \tan u)\, u'$
- $\dfrac{d}{dx}\left[\, \csc u \,\right] = -(\csc u\ \cot u)\, u'$
- $\dfrac{d}{dx}\left[\, \cot u \,\right] = -(\csc^2 u)\, u'$

Exponentials and Logarithms

Assume u is a differentiable function of x.

- $\dfrac{d}{dx}\left[\, e^{ax} \,\right] = ae^{ax}$
- $\dfrac{d}{dx}\left[\, e^{u} \,\right] = e^{u}\, u'$
- $\dfrac{d}{dx}\left[\, \ln x \,\right] = \dfrac{1}{x}$ for $x > 0$
- $\dfrac{d}{dx}\left[\, \ln u \,\right] = \dfrac{1}{u}\, u'$ for $u > 0$

Appendix C

A Quick Guide to MATLAB®

In this course we will be using the software package MATLAB® . The most recent version can be purchased directly from the MATLAB® web site: http://www.mathworks.com/academia/student_version/index.html

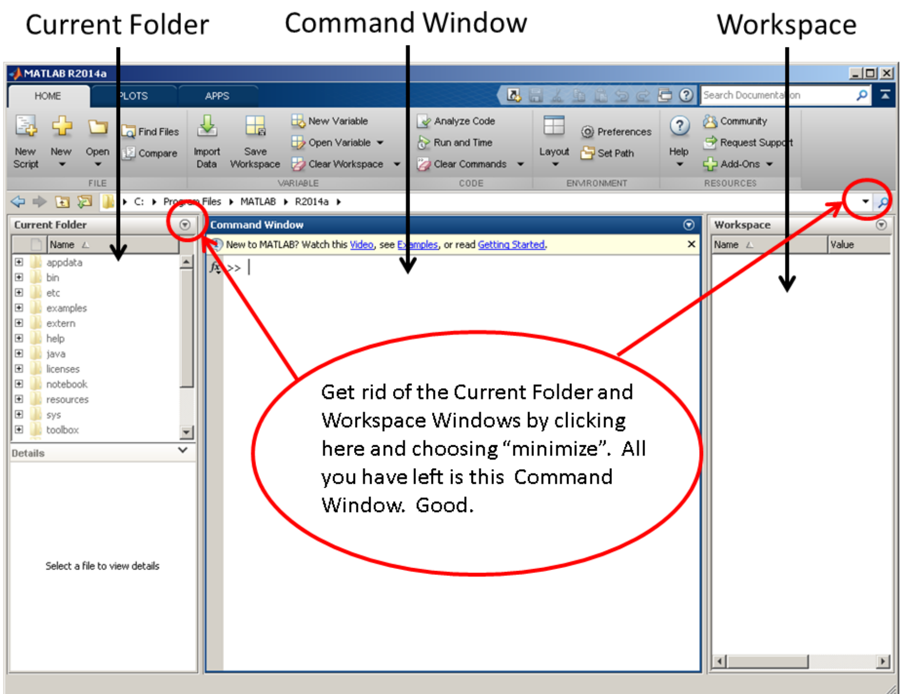
MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. We will use MATLAB® for it's ability to easily perform matrix and vector operations as well as it's exceptional ease of graphing and animating objects in 2 and 3 dimensions. This Quick Guide to MATLAB® is meant to demonstrate some of the more popular tasks that we will be performing with MATLAB® .

Contents

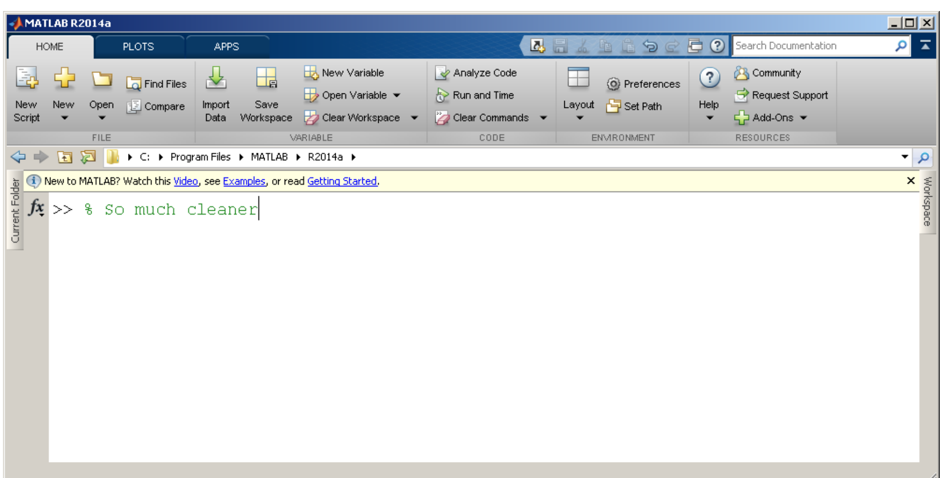
All of the files demonstrated in these pages can be found in the program repository.

C.1 Running MATLAB®

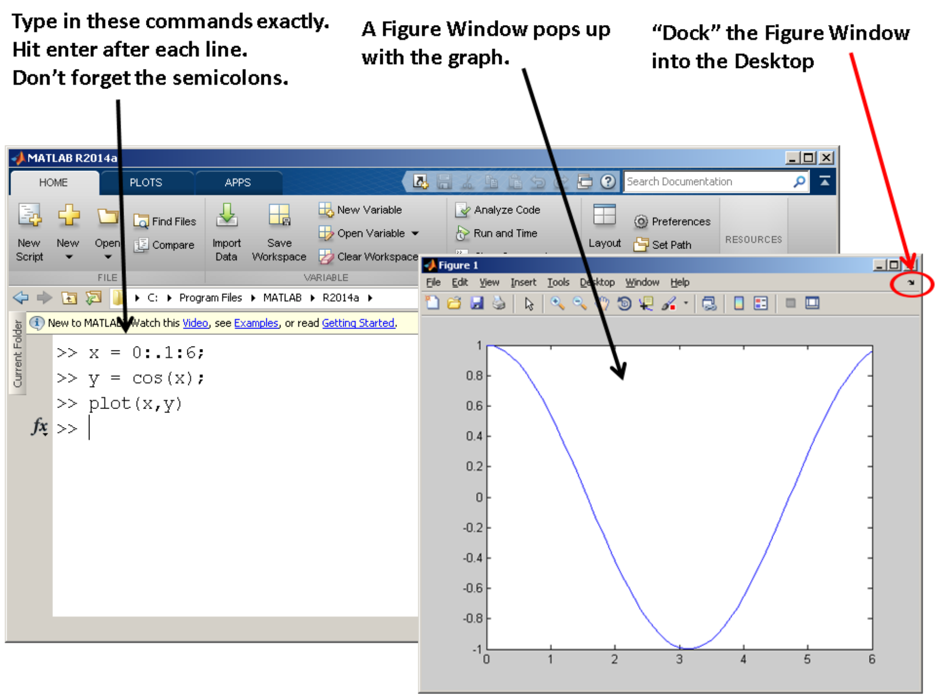
When you first start MATLAB® the desktop will open. The original desktop has too much going on for my preference. You can clean this up by first removing all windows except the Command Window.



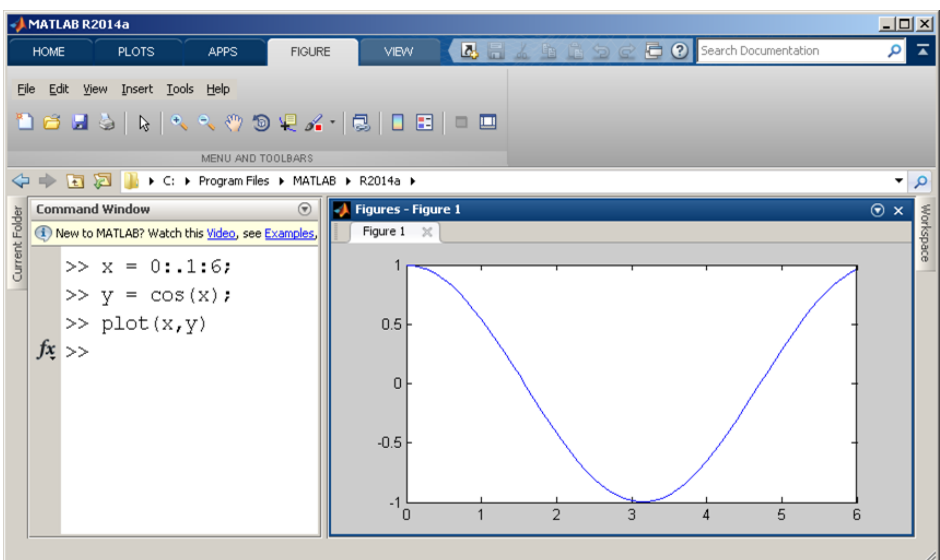
The Desktop Becomes



Type in the commands below. The first line `x = 0:.1:6` creates an array of x values starting at zero and incrementing by 0.1 up to 6. Don't forget the semicolons.



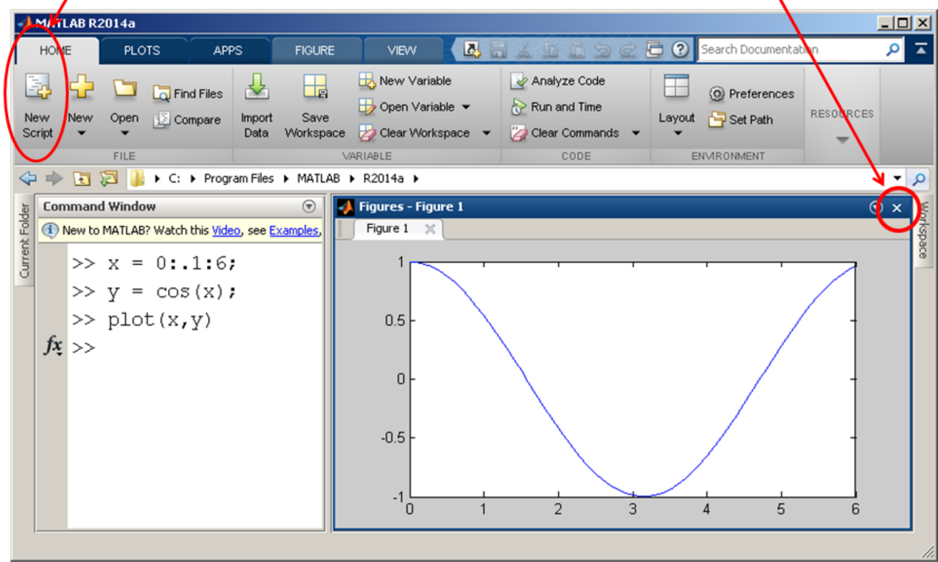
The Desktop Becomes



1) Click on the 'Home' Tab
2) Click on 'New Script'

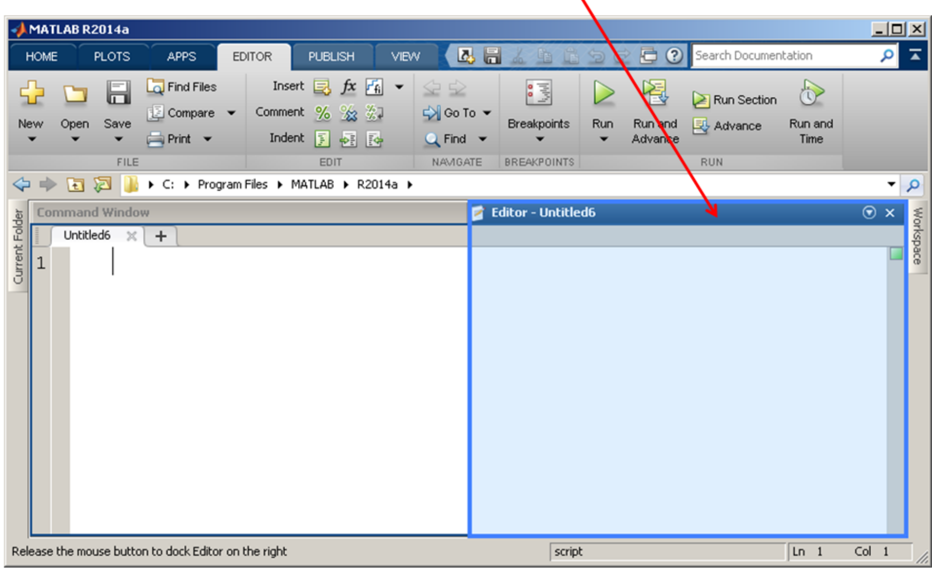
3) An editor window will open. This is where we write code.

4) Close the Figure window for now.

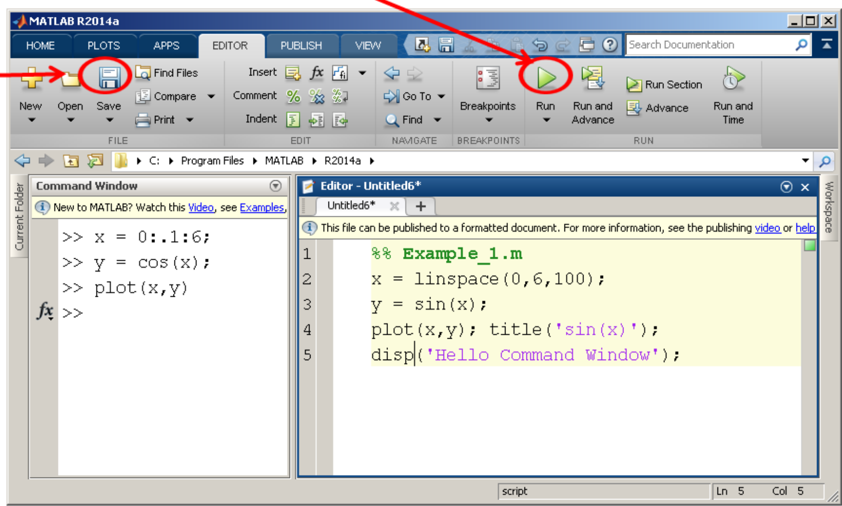


Initially, the Command Window and Editor Window are tabbed. We want them side-by-side.

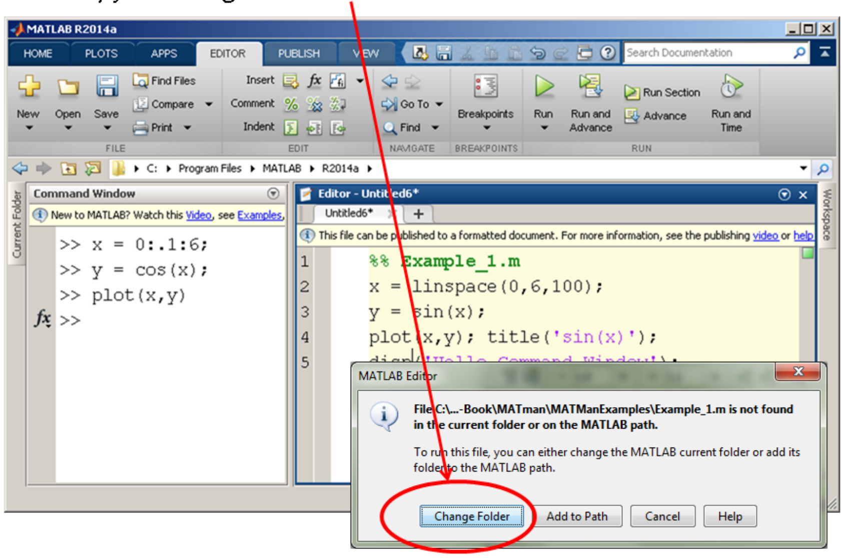
Left-click and hold on the blue bar at the top of the Editor Window. Then drag the Editor Window down and right until the right-half of the window is shaded. Release the left-click.



1) In the editor, type: %% Example_1.m (this will be the name of the file)
2) Type in the following lines which will be executed later.
3) The disp command prints to the Command Window.
4) Save it as Example_1.m to a folder you can find later.
5) Hit the green arrow to run it.



This file is probably not saved in the current folder where MATLAB is running.
You can either change the current folder to that containing the file or
you can add a path to the folder containing the file.
I usually just Change Folder

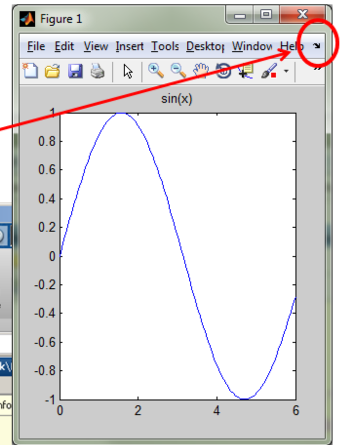


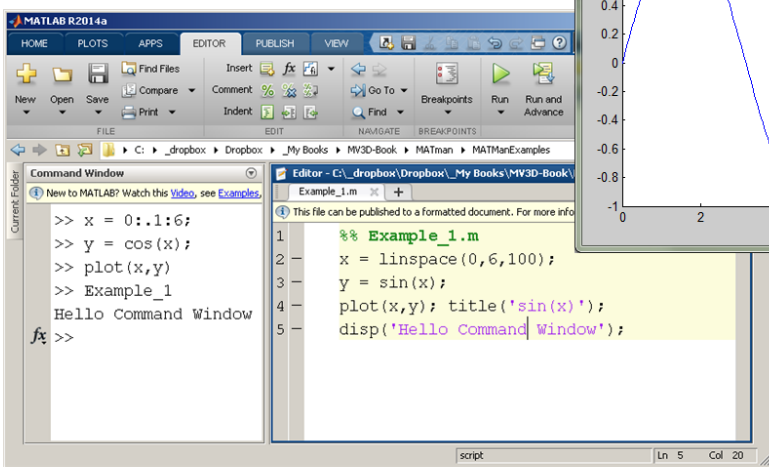
The file is then executed line-by-line.

A Figure window opens with the graph.

Your message is displayed in the Command Window

Now, "Dock" the Figure Window

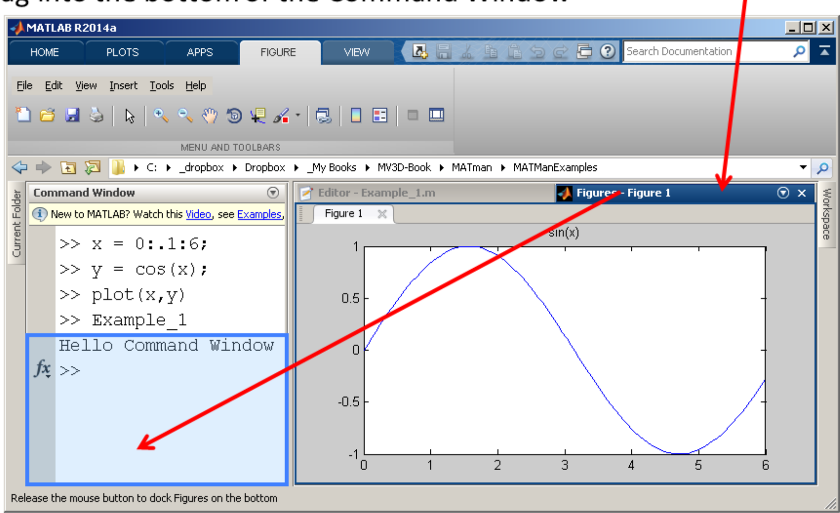


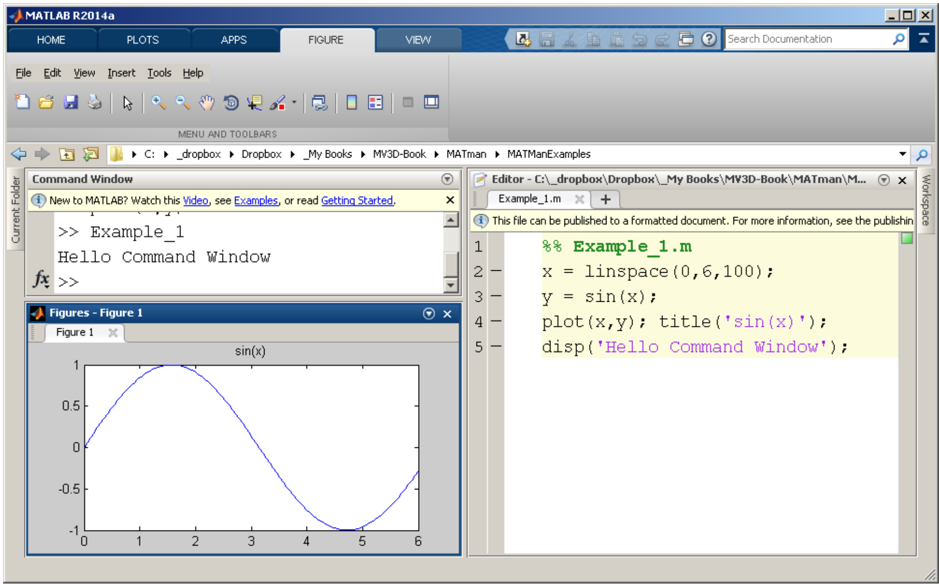


The figure is now docked but it is tabbed with the Editor. I don't like this.

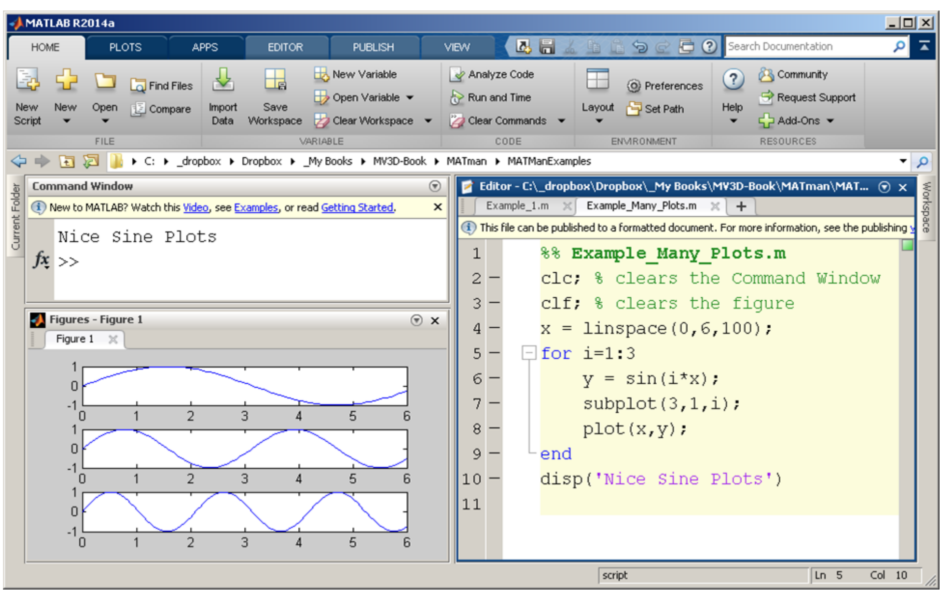
Grab (right-click and hold) the figure window by the blue bar.

Drag into the bottom of the Command Window





- Resize and move things around to your liking.
- The next time you open MATLAB® it will remember how you like it.
- Play around with the code. Hit the green arrow to run it.
- Save it just like any other file. It must have a .m extension.



C.2 Creating your own functions

Often we will want to create functions that take input and return output. These *functions* can be nested within a program or saved in a separate file called a function file.

- **Function Files (best - most versatile)**

  Here, we create a file that acts as a function. These are nice because the function can be composed of many parts, it can return numerous variables, or none at all. A function file can be used to create graphs with various features described by the input variables.

  Important Stuff:

  1. The function file must be in your working directory or a path to it must be created.
  2. The first line must be `function [output variable(s)] = Function_Name(input variable(s))`
  3. The file must be named `Function_Name.m`.

  ```
  Usage:

  function [output variable(s)] = Function_Name(input variable(s))
          function operations;
          output variables;
          ⋮
          other stuff;
  ```

- **Anonymous Functions (good - less versatile but easy)**

  Here you create the function directly in the program - this is nice. The problem is the function must consist of only one line of code. Multi-part functions with a `for` loop (for example) can't be used. An anonymous function can return more than one variable (good).

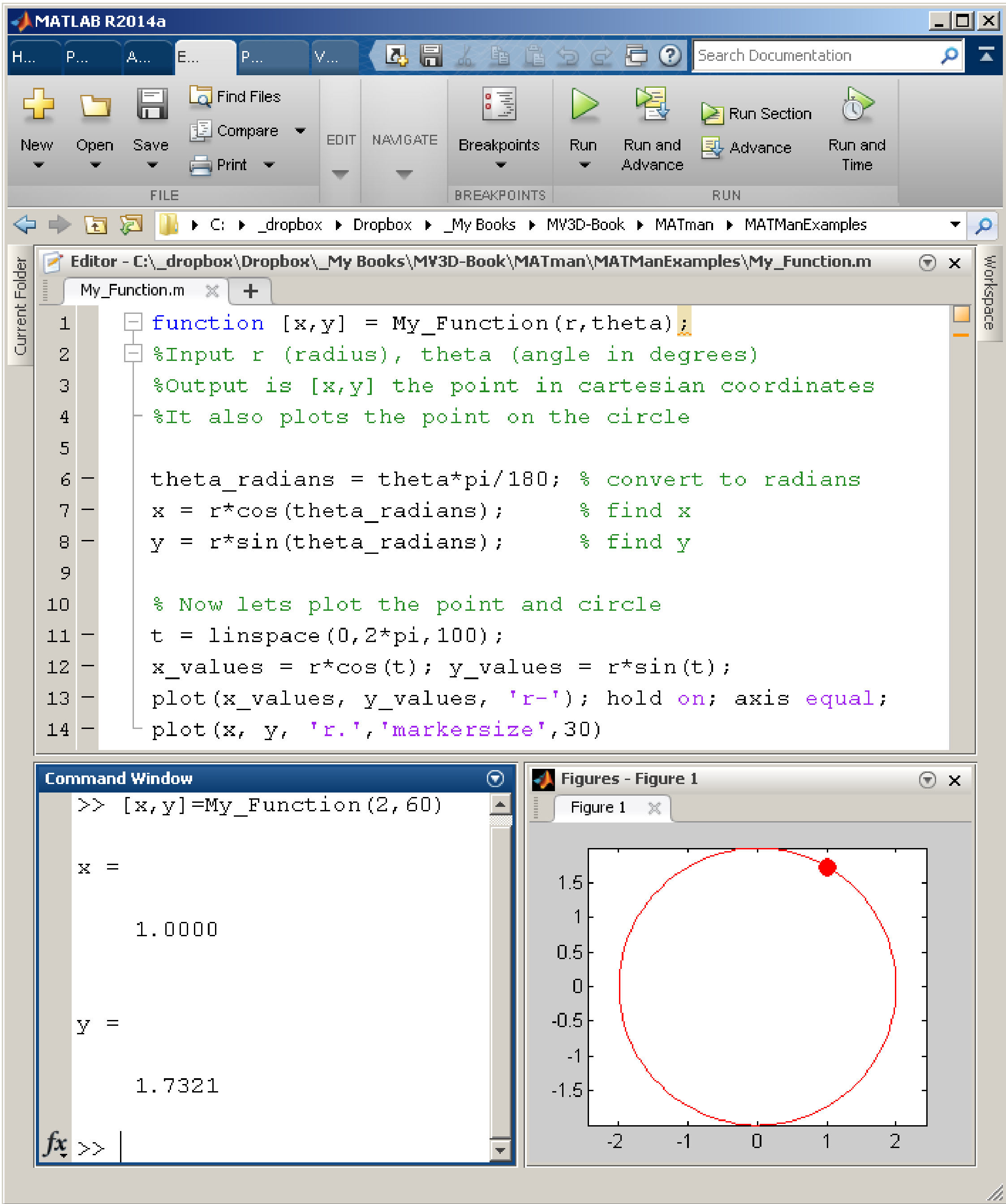
  ```
  Usage:
  function_name = @(input variables)[function output variable(s)]
  ```

- **Inline Functions (bad - not supported in future versions of MATLAB®)**

  This type of function is being phased out. It is being described here only because these functions can show up in older code. They should be replaced with anonymous functions. This type of function can only return one variable but can accept numerous input variables.

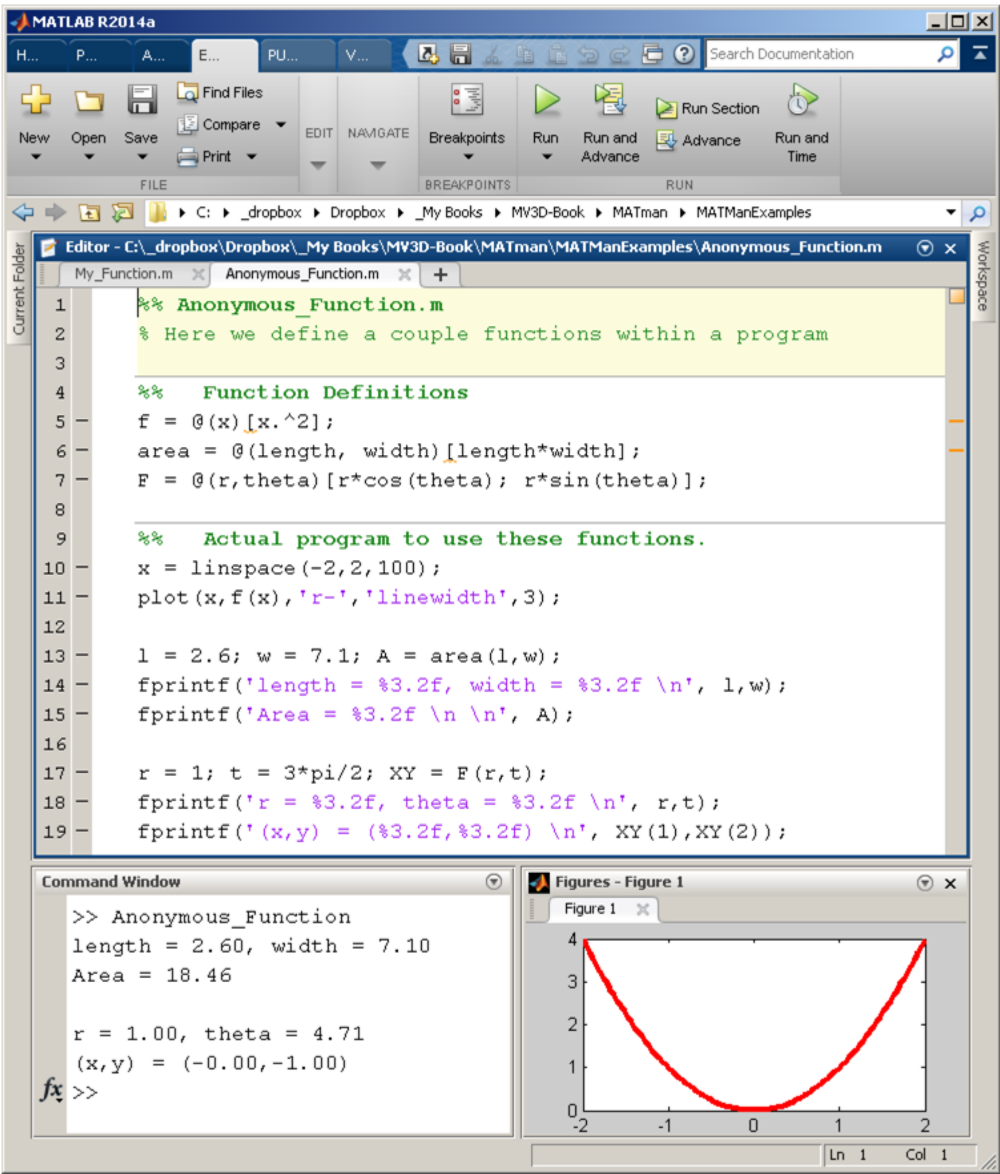
  ```
  Usage:
  function_name = inline('operation', 'input variable 1', 'input variable 2')
  ```

Function Files: Here, we create a file that acts as a function. We can call this function from the command window, by typing `[x,y] = My_Function(2,60)` (as below) or call it from another program file.



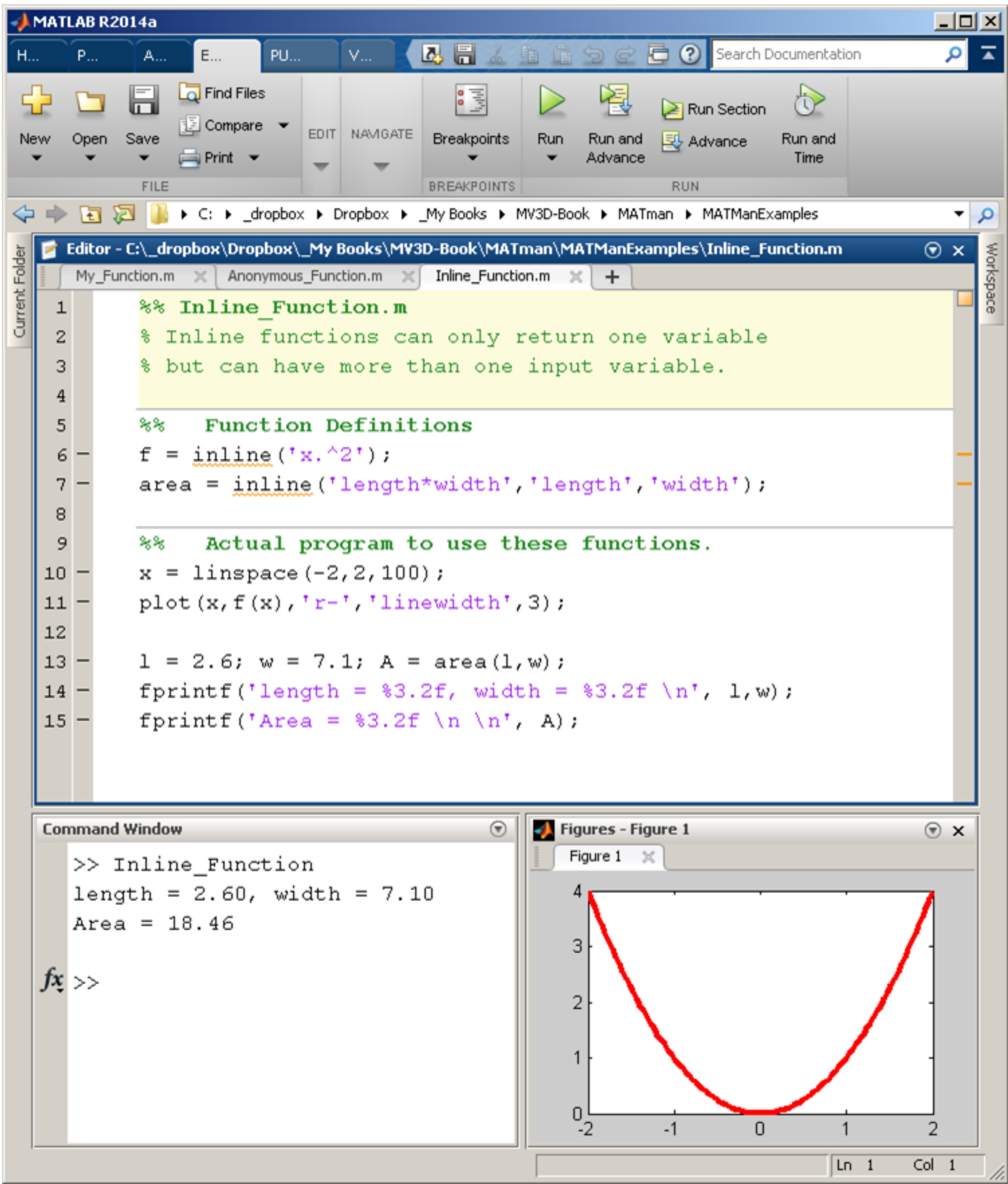
Anonymous Function: Versatile and Easy

MATLAB® Notation	Math Notation	Description
`f = @(x)[x.^2]`	$f(x) = x^2$	One input (x), One output (f)
`area = @(l,w)[l * w]`	$\text{area}(l, w) = lw$	Two inputs (l,w), One output (area)
`F = @(r,t)[r*cos(t), r*sin(t)]`	$\mathbf{F}(r, t) = [r\cos(t), r\sin(t)]$	Two inputs (r,t), Two outputs (x,y)



<u>Inline Functions</u>: Bad - Not supported in future versions of MATLAB® .
These are described here only because they show up in older code.

MATLAB® Inline Function Notation	Math Notation
`f = inline('x.^2')`	$f(x) = x^2$
`area = inline('length * width', 'length','width')`	$area(l, w) = lw$



C.3 Graphing with MATLAB®

Here you find some examples of the built in graphing capabilities of MATLAB® . I'll demonstrate some simple ones and others with bells and whistles. For more details check the help browser (click on `help` or `?` in the toolbar) and search. The internet is well populated with MATLAB® help. I can usually find what I need faster by typing my question in any internet search engine.

Plotting y = f(x)

```
x = linspace(0,4,100);
y = cos(x);
plot(x,y, optional list of specifications)
```

Parametric Curves in 2D: x = f(t) and y = g(t)

```
t = linspace(0,2 * pi, 100);
x = cos(t);
y = sin(t);
plot(x,y, optional list of specifications)
```

Animations

```
for i = 1:n,
   ballplot = plot(x,y,..)
   update x and y
   pause(.01)
   delete(ballplot)
end
```

Plotting 3D Surfaces: z = f(x,y)

```
x1 = linspace(a, b, n);
y1 = linspace(c, d, m);
[x2, y2]= meshgrid(x1,y1)
z = f(x2,y2);
surf(x2, y2, z, optional list of specifications)
```

Parametric Curves in 3D: x = f(t), y = g(t), z = h(t)

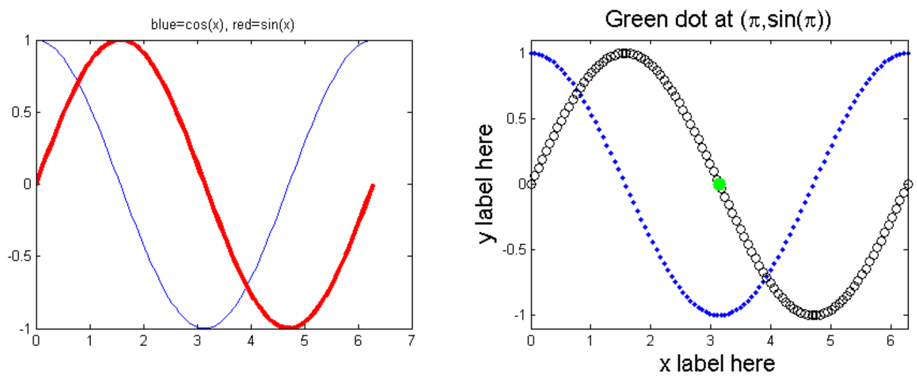
```
t = linspace(a,b,n);
x = f(t);
y = g(t);
z = h(t);
plot3(x,y,z, optional list of specifications)
```

Plotting y = f(x)

MATLAB® code: `Example_Plot.m` found in the program repository.

```
%% Example_Plot.m
clc; clf; % clears the command window (clc) and figures (clf)

%% Make the x and y vectors
x =  linspace(0,2*pi,100); % create a vector of x-values
% this vector starts at 0, ends at 2*pi, and has 100 terms
y1 = cos(x);     % define a vector of y-values called y1
y2 = sin(x);     % define a vector of y-values called y2

%% Start Plotting
subplot(1,2,1) % Create the first in a 1x2 array of plots
  % subplot is used to create different graphs in one figure
plot(x,y1);    % minimum requirements
hold on;       % Don't erase the graph
plot(x,y2,'r','linewidth',3);  % 'r' = red and thicker
title('blue=cos(x), red=sin(x)')

subplot(1,2,2) % Create the second in 1x2 array of plots
plot(x,y1,'b.');  % 'b.' = blue dots.
hold on;
plot(x,y2,'ko');  % black (k) circles
plot([pi],[sin(pi)],'g.','markersize',30);
title('Green dot at (\pi,sin(\pi))','fontsize',16)
xlabel('x label here','fontsize',16)
ylabel('y label here','fontsize',16)
axis([0,2*pi,-1.1,1.1]); % bounds [xmin, xmax, ymin, ymax]
```

Output:

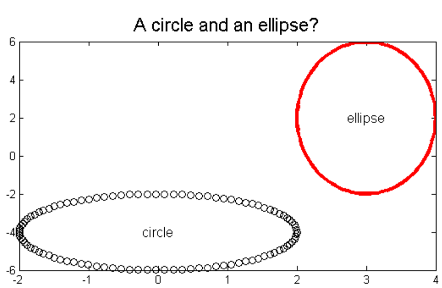


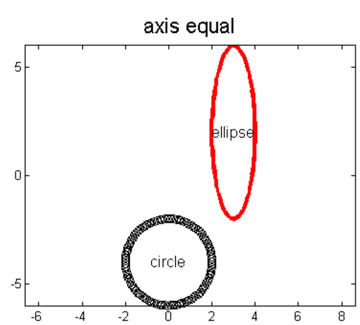
Parametric Curves in 2D: x = f(t) and y = g(t)

MATLAB® code: `Example_Parametric_Plot.m` found in the program repository.

```
%% Example_ParametricPlot.m
clc; clf; % clear the command window (clc); and the figure (clf)

%% The parameter (t) will go from zero to  2 pi.
t = linspace(0,2*pi,100);

%% A circle at (0,-4) with radius 2
x1 = 0 + 2*cos(t);  % define a vector of x-values called x1
y1 = -4 + 2*sin(t);  % define a vector of y-values called y1

%% An ellipse at (3,2) with x-radius = 1 and y-radius = 4
x2 = 3 + 1*cos(t);  % define a vector of x-values called x1
y2 = 2 + 4*sin(t);  % define a vector of y-values called y1

%% First Plot: Circle does not look like a circle
figure(1)
plot(x1,y1,'ko'); hold on; % plot the circle; don't erase;
text(0,-4,'circle','HorizontalAlignment','center','fontsize',12)
plot(x2,y2,'-r','linewidth',3) % plot the ellipse
text(3,2,'ellipse','HorizontalAlignment','center','fontsize',12)
title('A circle and an ellipse?','fontsize',16) % title

%% Second plot has equal scaling with axis equal;
figure(2)
plot(x1,y1,'ko'); hold on; % plot the circle; hold it
text(0,-4,'circle','HorizontalAlignment','center','fontsize',12)
plot(x2,y2,'-r','linewidth',3) % plot the ellipse
text(3,2,'ellipse','HorizontalAlignment','center','fontsize',12)
title('axis equal','fontsize',16);
axis equal;
```

Output:



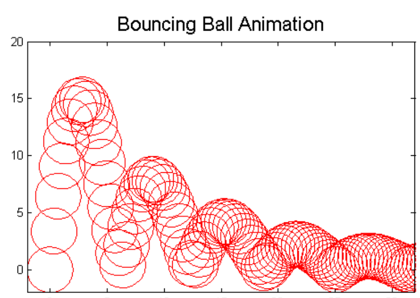


Animations

MATLAB® code: `Bouncing_Ball_Animation.m` found in the program repository.

```
%%   This is Bouncing_Ball_Animation.m
clc; clf; clear; % clear command window (clc), figure (clf), and all variables (clear)

%% The path the ball will follow:
path = @(x)[20*exp(-.1*x) .* abs(sin(.5*x))];

%% Initial Set-Up
n = 300;                % number of frames.
x = linspace(0,35,n);   % x values of the path.
y = path(x);            % y values of the path.

t = linspace(0,2*pi,100);   % the parameter t for making the ball.
original_xball = 2*cos(t);  % the x-values of the ball graph.
original_yball = 2*sin(t);  % the y-values of the ball graph.

%% Make original plot
pathplot = plot(x,y,'-k'); hold on;                       % plot the path
ballplot = plot(original_xball,original_yball,'-r');      % plot original ball
title('Bouncing Ball - Click to Start','fontsize',16);
axis equal; axis([-2,37,-2,20]);

% Initiate animation
waitforbuttonpress;                                   % press any button to continue
title('Bouncing Ball Animation','fontsize',16)
delete(pathplot);                                     % delete the path plot

%% Aimation Starts Here
for i=1:n,            % begin for loop
  xball = original_xball+x(i);    % move ball x-values along path.
  yball = original_yball+y(i);    % move ball y-values along path.
  delete(ballplot);               % delete old ballplot
  ballplot = plot(xball, yball, '-r');  % make new ball plot
  pause(.01);                     % slows it down a little.
end                               % end the animation loop.
```

Output:

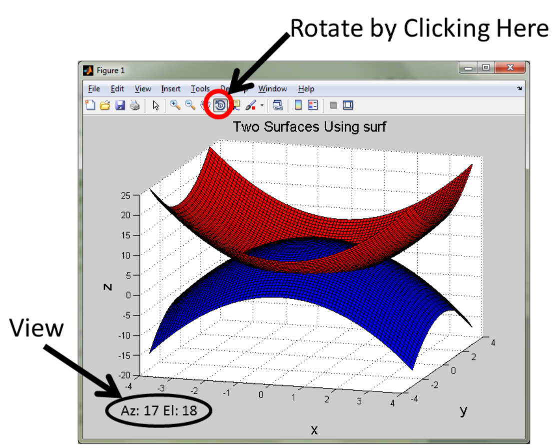


Plotting 3D Surfaces: z = f(x,y)

MATLAB® code: `Example_Surf.m` found in the program repository.

```
%%This is Example_Surf.m
% Uses the surf command to plot surfaces in 3D.
clc; clf; clear; % clear command window (clc), figure (clf), and all variables (clear)

%% create the x,y,z values
xvec = -4:.1:4;  % create the vector of x-values
yvec = -3:.1:3;  % create the vector of y-value
[x y] = meshgrid(xvec,yvec);  % create a full array of x & y
z1 = x.^2 + y.^2;  % define z1
z2 = 9-(x.^2 + y.^2);  % define z2

%% plot'em
surf(x, y, z1,'facecol','red');  % plot the surface
hold on; %don't erase the first graph
surf(x, y, z2,'facecol','blue'); % plot the surface in blue
title('Two Surfaces Using surf','fontsize',16)
xlabel('x','fontsize',16);
ylabel('y','fontsize',16);
zlabel('z','fontsize',16);
view(17,18);  % view (Az,El) found by rotating the graph.
```

- Rotate the graph.
- Observe the Az and El numbers in the lower left
- use `view(az,el)` to replicate the desired viewing angle.

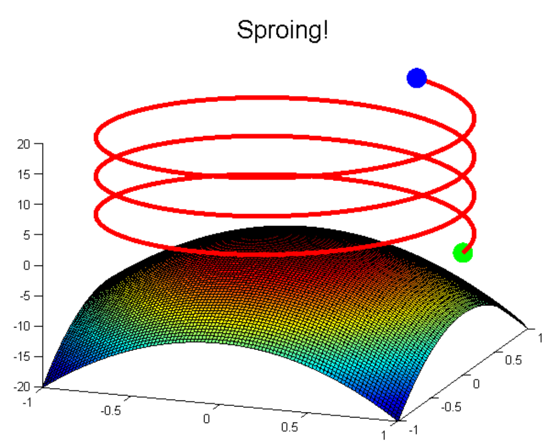


Parametric Curves in 3D: $x = f(t)$, $y = g(t)$, $z = h(t)$

MATLAB® code: `Example_Plot3.m` found in the program repository.

```
%% This is Example_Plot3.m. It plots curves, points, and a surface in 3D.
clc; clf; clear; % clear command window (clc), figure (clf), and all variables (clear)

%% First, plot a parametric curve in 3D using the standard form of plot3.
t = linspace(0,20,200);  % this is the parameter
x = cos(t); % this defines the x-values on the curve.
y = sin(t); % this defines the corresponding y-values
z = t;  % and this defines the corresponding z-values.
plot3(x,y,z,'r-','linewidth',3);    % Plot the Spring!
hold on;

%% Let's plot a couple points
plot3(cos(0),sin(0),0,'g.','markersize',50);        % green dot (bottom of spring)
plot3(cos(20),sin(20), 20 ,'b.','markersize',50);   % blue dot (top of spring)

%% How about a little surface in there
x1 = linspace(-1,1,100);  % create the vector of x-values for the surface plot
y1 = linspace(-1,1,100);  % create the vector of y-values for the surface plot
[x2,y2]=meshgrid(x1,y1);  % create a full array of x & y values for the surface plot
z = -10*(x2.^2 + y2.^2);  % define the z-values for the surface plot
surf(x2,y2,z);            % plot the surface
view(20,22)               % view(Az, El) found using the rotation button on the figure
title('Sproing!','fontsize',20);
```

Output:



C.4 Input and Output with the Command Window

While a program is running there are many situations where you will want to get input from the Command Window or send output to the Command Window. Input is achieved through the `input` command and output is achieved with a variety of print commands. The next two pages give examples of each.

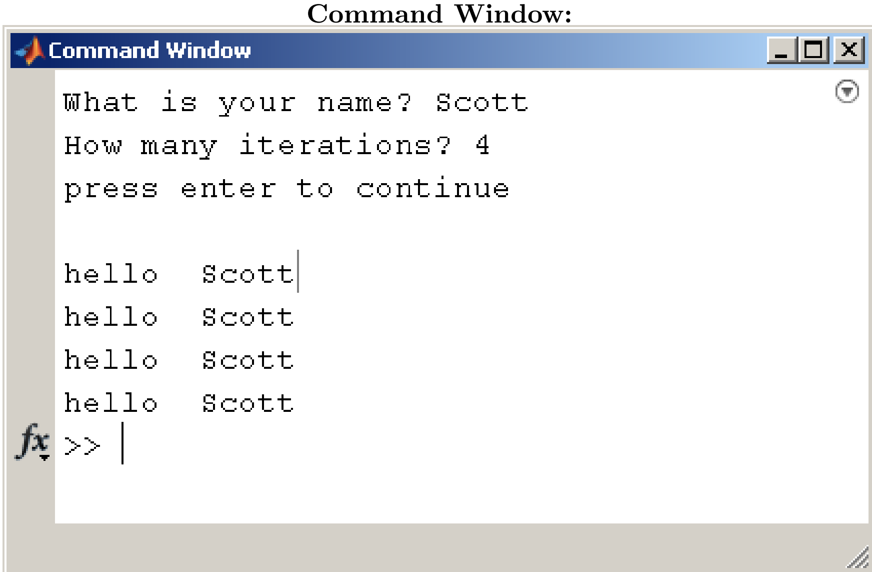
Input from the Command Window

- The `input` command is used to get numbers or strings from the command window while a program is running. The default value of `input` is a real number but a string can be accepted with the appropriate syntax as demonstrated in the code below.

MATLAB® code: `Example_Input_Command_Window.m` found in the program repository.

```
%% Example_Input_Command_Window.m
% It gets input from the command window using the input command.
clc; % clear the command window

name = input('What is your name? ','s');
n=input('How many iterations? ');
input('press enter to continue \n');
for i=1:1:n,
  fprintf('hello  ')
  disp(name)
end
```

Command Window:



Output to the Command Window

- The `disp` command is used to send simple strings to the command window.
- The `sprintf` command is used to create strings with numerical terms.
- The `fprintf` command works like `sprintf` except the results are printed to the command window.

MATLAB® code: `Example_Output_Command_Window.m` found in the program repository.

```
%% This is Example_Output_Command_Window.m
%  This demonstrates how to send output to the commmand window.
%  We use 'disp', 'fprintf', and 'sprintf'.
clc; clear; % clear command window(clc) and all variables (clear)

%% some stuff we will print
x1 = pi;                              % define variable x1
M = [ 1 2; 51.76 4];                   % define a Matrix M
text1 = sprintf('x1 = %1.5f \n',x1);
    % Above creates a string with a 5 decimal float (x1) inserted in the string.


%% The disp command is the easiest & fastest way to print something
disp('The disp command prints strings and variables to the command window')
disp(text1); % display the string variable text1
disp(M);     % display the matrix M


%% The fprintf command prints strings and numbers
fprintf('The printf command can be use like the disp command ')
fprintf('but \n you need backslash n for a new line \n')
fprintf('It is good for printing numbers as strings such as ')
fprintf('\n If x = %1.5f then sin(x) = %1.2f \n',x1,sin(x1))
```

Results get printed to the command window.

```
Command Window
The disp command prints strings and variables to the command window
x1 = 3.14159

    1.0000    2.0000
   51.7600    4.0000

The printf command can be use like the disp command but
 you need backslash n for a new line
It is good for printing numbers as strings such as
 If x = 3.14159 then sin(x) = 0.00
>> 
```

C.5 Input and Output with a Figure

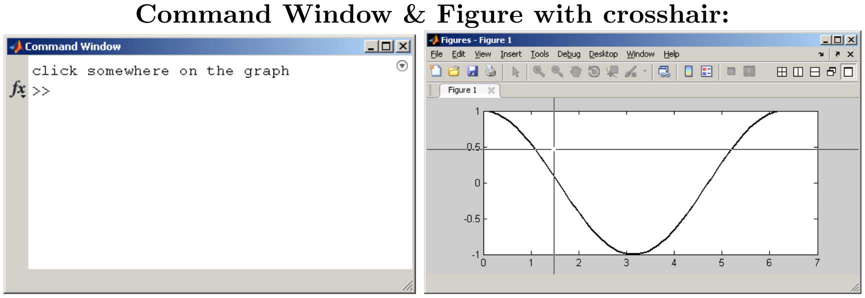
Input from a Graph or Figure

- The `ginput` command is used to get a point from a mouse click on a graph.

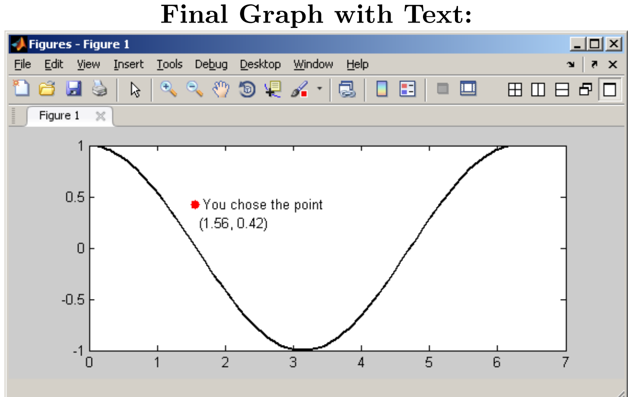
MATLAB® code: `Example_Input_Graph.m` found in the program repository.

```
%% Example_Input_Graph.m
%  Getting input from a graph or figure using the locate command
clc; clf; % clear command window and figure;

%%  Plot a curve and get a point from the user
x_vector = linspace(0,2*pi,100);  % create a vector of x-values
y_vector = cos(x_vector);          % create a vector of y=cos(x) values
plot(x_vector,y_vector,'-k','linewidth',2);   % plot x and y
hold on;                                      % do not erase the graph
disp('click somewhere on the graph')          % message to command window
[x,y] = ginput(1);                            % choose one point on graph
plot(x,y,'r.','markersize',20)                % plot that point on the graph

%% Print a message to the figure
text(x+.1,y,'You chose the point');         % text(x-location, y-location, 'text')
text1 = sprintf('\n (%1.2f, %1.2f)',x,y);   % create text with floats in it
text(x,y-.1,text1);                         % text(x-location, y-location, text)
```

Command Window & Figure with crosshair:



Final Graph with Text:



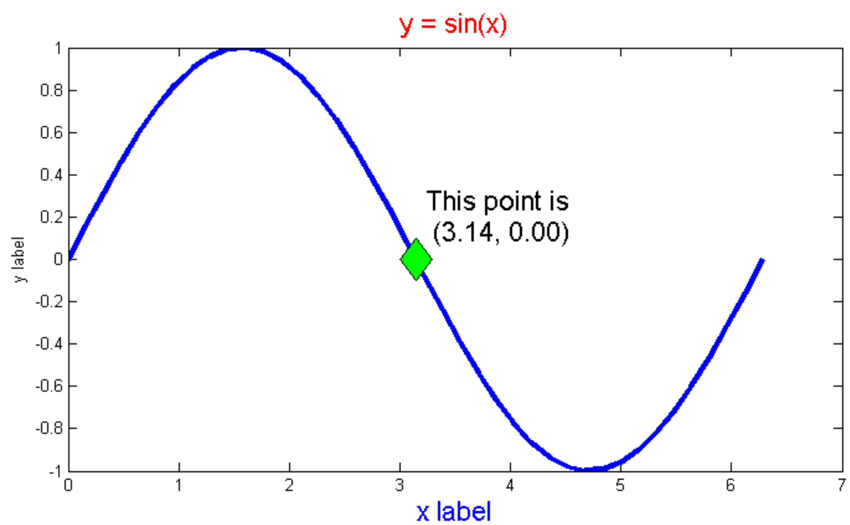
Output to a Graph

- The `text` command is used to place text to a specific location in the graph.
- The `sprintf` command is used to create strings with numerical terms.

MATLAB® code: `Example_Output_Graph.m` found in the program repository.

```
%% This is Example_Output_Graph.m
% This demonstrates how you can send output to a graph.
clc; clf; clear; % clears console, figure, and variables.

%% we will plot y = sin(x) at 40 evenly located x-values
x_vector = linspace(0,2*pi,40);  % x = 0 to 2 pi with 40 points.
y_vector = sin(x_vector);        % y = sin(x)
x1 = pi;    % an x-value

%% Plot some stuff with a message in the graph
plot(x_vector,y_vector,'linewidth',3); hold on;
plot([x1],[sin(x1)],'kd','markersize',20,'markerface','green')
    % Above: plot the point (x1,sin(x1)) with a green diamond.
 title('y = sin(x)', 'fontsize',16,'color','red');
 xlabel('x label','fontsize',16,'color','blue');
 ylabel('y label');
text_stuff = sprintf('This point is \n (%1.2f, %1.2f)',x1,sin(x1));
text(x1+.1,sin(x1)+.2,text_stuff,'fontsize',16);
```

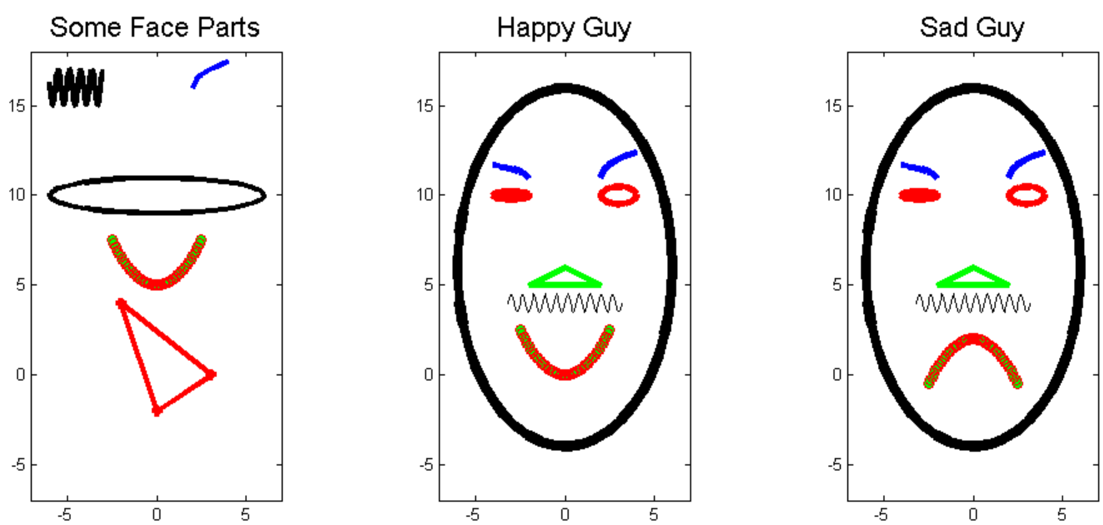
Resulting Graph with Text:



C.6 Assignment

Create a Face.
Create a face using the plotting commands introduced in this chapter.
You can start with the `My_Face_Starter.m` file found in the program repository.

- Minimum Requirements
  - The face must have a mouth, nose, and two eyes.
  - The features must be distinguished by different plotting styles.
  - It must be significantly different from the one below.
  - It must contain at least 3 ellipses and 1 trig (sin or cos) function.
  - It must contain at least 1 polygon (like a triangle).
  - There can be no error messages upon running the code.
- Recommendations
  - Additional unique facial features.
  - Prompt user for types of facial features from the command window. The smile/frown option from the starter code does not count.
- Additional Ideas
  - Prompt user for facial feature location by clicking on the graph.
  - Animate the face. The axes must remain constant during the entire animation and it must stop on its own (no infinite loops).



Getting Started: This code generates the first graph on the previous page. Download it from the program repository. Run it. Determine what each line of code does to make the various graphs in the figure.

MATLAB® code: `My_Face_Starter.m` found in the program repository.

```
%% This is My_Face_Starter.m
clc; clf; clear;  %clears command window (clc), figure (clf), variables (clear)

%% Get numerical input from command window. 1=smile, 0=frown.
smile = input('input smile (1) or frown (0)');

%% Make a smile (or frown) and plot it
mouth_X = linspace(-2.5,2.5,50);  % x-values for the mouth
mouth_parabola = .4*mouth_X.^2;   % makes a parabola that opens up
                                  % the .^2 squares each term in mouth_X
if smile == 1                     % if smile is chosen
    mouth_Y = mouth_parabola + 5; % move parabola up 5 units.
else                              % otherwise
    mouth_Y = -mouth_parabola + 5;  % multiply by -1 (open down)
end                                 % and then move parabola up 5 units.
plot(mouth_X,mouth_Y,'ro','markerface','green'); % plot the mouth
hold on;  % do not erase the graph

%% Make a triangle and plot it
triangle_X = [-2,3,0,-2];   % x-vertices of the triangle
triangle_Y=[4,0,-2,4];      % y-vertices of the triangle
plot(triangle_X,triangle_Y,'-r*','linewidth',3);  % plot it

%% Make hair and plot it
hair_X=linspace(-6,-3,100);  %x-values for the hair
hair_Y=sin(10*hair_X) + 16;  %y-values for the hair
plot(hair_X,hair_Y,'k','linewidth',3)  % plot the hair

%% Make an eyebrow and plot it
eyebrowX = linspace(2,4,10);     % x-values of the eybrow
eyebrowY = 16+sqrt(eyebrowX-2);  % y-values of the eyebrow
plot(eyebrowX,eyebrowY,'linewidth',3);  % plot the eyebrow

%% Make a face (ellipse) and plot it
face_T = linspace(0,2*pi,50);  % parameter t-values for the ellipse
face_X = 6*cos(face_T);        % x-values of the ellipse
face_Y = 10 + sin(face_T);     % y-values of the ellipse
plot(face_X,face_Y,'k','linewidth',3); % plot the ellipse

%% Some graphics directives
title('Some Face Parts','fontsize',16); % give the graph a title
axis equal;           % Makes circles look like circles!
axis([-7,7,-7,18]);   % graph bounds [xmin,xmax,ymin,ymax]
```

Some Graphing Tricks

- If you want to draw an ellipse (or circle for that matter). Do so in parametric form;

  $$x(t) = x_o + r_x \cos t, \qquad y(t) = y_o + r_y \sin t, \qquad \text{for} \quad t \in [0, 2\pi].$$

  then just `plot(x,y)`. This ellipse has center at (x_o, y_o), x-radius of r_x, and y-radius of r_y.

- Suppose you graph $y = f(x)$, then
  - $y = f(x) + a$ moves the graph up a units.
  - $y = f(x - a)$ moves the graph to the right a units.
  - $y = -f(x)$ flips the graph around the x-axis.
  - $y = f(-x)$ flips the graph around the y-axis.
  - $y = a \cdot f(x)$ scales (makes it taller or shorter) the graph by a units.

- If you want your graph to have specific bounds use

  `axis([`x_{min}`, `x_{max}`, `y_{min}`, `y_{max}`])`

- You should set `axis equal;` so that circles look like circles.

- Order Matters: If you want to determine your own axes bounds and scale them equally you must use these commands in this order:
  1. `axis equal:`
  2. `axis([`x_{min}`, `x_{max}`, `y_{min}`, `y_{max}`])`

  If you do these in the opposite order MATLAB® will first set the bounds and then, upon scaling them equally, will reset the axis limits happily violating the ones you had just set.

Detailed Solutions to Worksheets

A few of the sections have an associated worksheet. Detailed solutions are found here.

Chapter 1.1 Worksheet - Solutions

Use Gaussian elimination with back-substitution to determine if the system of equations is consistent (has at least one solution). If it is consistent, solve for the variables. If you get infinitely many solutions, give the **general solution** in terms of a parameter (t) and give one **particular solution**.

1.
$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

Expressing this system in augmented matrix form and performing Gaussian Elimination:

$$\left[\begin{array}{ccc|c} 1 & 0 & -3 & 8 \\ 2 & 2 & 9 & 7 \\ 0 & 1 & 5 & -2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & 8 \\ 0 & 2 & 15 & -9 \\ 0 & 1 & 5 & -2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & 8 \\ 0 & 1 & 5 & -2 \\ 0 & 2 & 15 & -9 \end{array}\right]$$
$$\sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & 8 \\ 0 & 1 & 5 & -2 \\ 0 & 0 & 5 & -5 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & 8 \\ 0 & 1 & 5 & -2 \\ 0 & 0 & 1 & -1 \end{array}\right]$$

Equation 3: $\mathbf{x_3 = -1}$.
Equation 2: $x_2 + 5x_3 = -2 \quad \rightarrow \quad x_2 + 5(-1) = -2 \quad \rightarrow \quad \mathbf{x_2 = 3}$.
Equation 1: $x_1 - 3x_3 = 8 \quad \rightarrow \quad x_1 - 3(-1) = 8 \quad \rightarrow \quad \mathbf{x_1 = 5}$.
The system is **consistent** with unique solution $(\mathbf{x_1, x_2, x_3}) = (\mathbf{5, 3, -1})$.

2.
$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

Expressing this system in augmented matrix form and performing Gaussian Elimination:

$$\left[\begin{array}{ccc|c} 0 & 1 & -4 & 8 \\ 2 & -3 & 2 & 1 \\ 5 & -8 & 7 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|c} 2 & -3 & 2 & 1 \\ 0 & 1 & -4 & 8 \\ 5 & -8 & 7 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & -3/2 & 1 & 1/2 \\ 0 & 1 & -4 & 8 \\ 5 & -8 & 7 & 1 \end{array}\right]$$
$$\sim \left[\begin{array}{ccc|c} 1 & -3/2 & 1 & 1/2 \\ 0 & 1 & -4 & 8 \\ 0 & -1/2 & 2 & -3/2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & -3/2 & 1 & 1/2 \\ 0 & 1 & -4 & 8 \\ 0 & 0 & 0 & 5/2 \end{array}\right]$$

Equation 3: $0x_1 + 0x_2 + 0x_3 = 5/2$.
There are no solutions to this equation and hence no solution to the system of equations.
Therefore, the system is **inconsistent**.

3.
$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 10 \\ x_1 + 5x_2 &= 6 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

Expressing this system in augmented matrix form and performing Gaussian Elimination:

$$\left[\begin{array}{ccc|c} 1 & 6 & 2 & 10 \\ 1 & 5 & 0 & 6 \\ 0 & 1 & 2 & 4 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 6 & 2 & 10 \\ 0 & -1 & -2 & -4 \\ 0 & 1 & 2 & 4 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 6 & 2 & 10 \\ 0 & 1 & 2 & 4 \\ 0 & 1 & 2 & 4 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 6 & 2 & 10 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Equation 3: $0x_1 + 0x_2 + 0x_3 = 0$.
We will get infinitely many solutions so the system is **consistent and dependent**.
We seek a general solution by setting x_3 equal to any real number.

Let $\mathbf{x_3 = t}$ (any real number)

From the second equation:
$x_2 + 2x_3 = 4$
$x_2 = 4 - 2x_3$
$\mathbf{x_2 = 4 - 2t}$

From the first equation:
$x_1 + 6x_2 + 2x_3 = 10$
$x_1 = 10 - 6x_2 - 2x_3$
$x_1 = 10 - 6(4 - 2t) - 2t$
$x_1 = 10 - 24 + 12t - 2t$
$\mathbf{x_1 = -14 + 10t}$

The **general solution** is given in the form

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -14 + 10t \\ 4 - 2t \\ t \end{bmatrix} \quad \text{for} \quad -\infty < t < \infty.$$

A **particular solution** is found by assigning any number to the parameter t. For example, if we set $t = 0$, a particular solution is $x_1 = -14$, $x_2 = 4$, and $x_3 = 0$.

Chapter 1.2 Worksheet - Solutions

1. **(MATLAB®)** Recall the following systems that you solved by hand in Section 1.1.

(a)
$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

(b)
$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

(c)
$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 10 \\ x_1 + 5x_2 &= 6 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

You should have found that (a) had a unique solution $[x_1, x_2, x_3] = [5, 3, -1]$, (b) (inconsistent) had no solutions, and (c) (dependent) had infinitely many solutions of the form $[x_1, x_2, x_3] = [6 - 5t, 2 - \frac{t}{2}, t]$.

Assignment: Use MATLAB® to solve these three systems of equations. You should run into problems. You will get warnings for parts (b) and (c). In part (b) MATLAB® actually gives you a solution that is not a solution and in part (c) MATLAB® does not return a solution.

(a)
Here, you set `A = [1 0 -3; 2 2 9; 0 1 5]` and `b = [8 7 -2]`.
Then you get the solution is `x = A\b = [5 3 -1]` with no error message.

(b)
Here, you set `A = [0 1 -4; 2 -3 2; 5 -8 7]` and `b = [8 1 1]`.
When you enter the command `A\b` you get a message.

```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 3.416071e-018.
```

However, it gives you a solution $(1.0e + 016) * [-1.1259, -0.9007, -0.2252]$. These are huge numbers because they are being multiplied by $\approx 10^{16}$. If you check this answer it is not really a solution.

(c)
Here, you set `A = [1 6 2; 1 5 0; 0 1 2]` and `b = [10 6 4]`.
When you enter the command `A\b` you get a message.

```
Warning: Matrix is singular to working precision.
```

It gives you a solution `[NaN, NaN, NaN]'`. Here, `NaN` means `not a number`. This will stop a program from running, which is good.

Observation: MATLAB® will give you a unique solution with no error message if a unique solution exists. However, if you have either no solutions or infinitely many solutions, MATLAB® will give you an error about the matrix being close to singular. It might give you a solution that is not a solution at all. It could also stop the process and not return any answer. The message about a singular matrix will be discussed later. For now, it is sufficient to say, if your coefficient matrix is singular - you got problems. The nice thing is, it doesn't happen that often. But when it happens things can turn ugly.

Chapter 1.3 Worksheet - Solutions

1. **(MATLAB®)** Recall the systems you solved by hand in Section 1.1 and with MATLAB® in 1.2:

(a)
$$\begin{aligned} x_1 - 3x_3 &= 8 \\ 2x_1 + 2x_2 + 9x_3 &= 7 \\ x_2 + 5x_3 &= -2 \end{aligned}$$

(b)
$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

(c)
$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 10 \\ x_1 + 5x_2 &= 6 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

You should have found that (a) had a unique solution $[x_1, x_2, x_3] = [5, 3, -1]$, (b) (inconsistent) had no solutions, and (c) (dependent) had infinitely many solutions of the form $[x_1, x_2, x_3] = [6 - 5t, 2 - \frac{t}{2}, t]$.

Assignment: Enter the coefficient matrices into MATLAB® , then find the determinants and inverses of these three matrices. You should run into problems.

For part (a), you should get a non-zero determinant and nice inverse matrix. For part (b), you should get something very close to zero for the determinant, and when you try to find its inverse you should get a warning and an inverse with crazy huge numbers. For part (c) you should get zero for the determinant and when you try to get its inverse you get a message that the matrix is singular and it returns a matrix of non-numbers.

(a) Here, you set `A = [1 0 -3; 2 2 9; 0 1 5]`.

Then `det(A) = -5` and `inv(A) =` $\begin{bmatrix} -0.2 & 0.6 & -1.2 \\ 3 & -1 & 3 \\ -0.4 & 0.2 & -0.4 \end{bmatrix}$. No error messages.

(b) Here, you set `A = [0 1 -4; 2 -3 2; 5 -8 7]`. Then `det(A) = 2.2204e-15`. This is close to zero (2.22 x10^{-15}) but is not zero as it should be. Since, as far as the software knows, this is not zero, it will try to get the inverse. When you enter `inv(A)`, you get the error message:

```
Warning :  matrix is close to singular or badly scaled.
Results may be inaccurate.  rcond = 3.416071e-018.
```

It then gives you an inverse with huge numbers.

(c) Here, you set `A = [1 6 2; 1 5 0; 0 1 2]`. Then `det(A) = 0` as it should and when you enter `inv(A)` you get the message:

```
Warning:  Matrix is singular to working precision.
```

It then returns a matrix with all terms `Inf` (which stands for infinity) - not a number. In this case, MATLAB® was able to tell you that the matrix has no inverse.

Observation: Inside of a machine the determinant of a matrix might not always give the exact value. This is a problem because if the determinant is zero, there is no inverse. If it is not zero, there is an inverse. Fortunately, when there is a problem you will usually be alerted by some type of message stating that the matrix is close to singular. It is always a good idea to figure out what caused these warnings.

Chapter 2.6 Worksheet - Solutions

Way back when solving systems of equations we came across the 3 x 3 system

$$\begin{aligned} x_2 - 4x_3 &= 8 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 8x_2 + 7x_3 &= 1 \end{aligned}$$

and we determined that there were no solutions. Allowing each equation to represent the graph of a plane, you can see that the planes are not parallel. Graph the three planes in such a way that you can ascertain why there is no point of intersection.

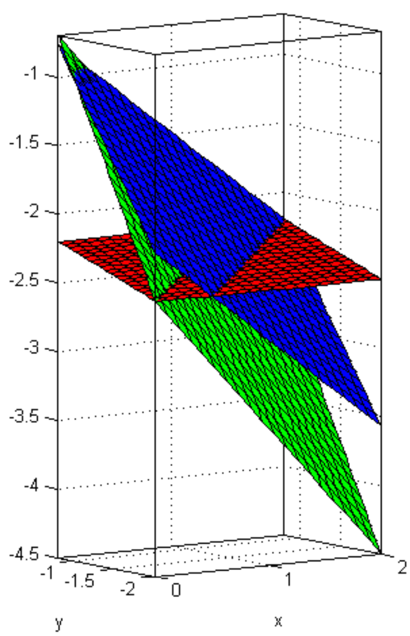
The normal vectors are

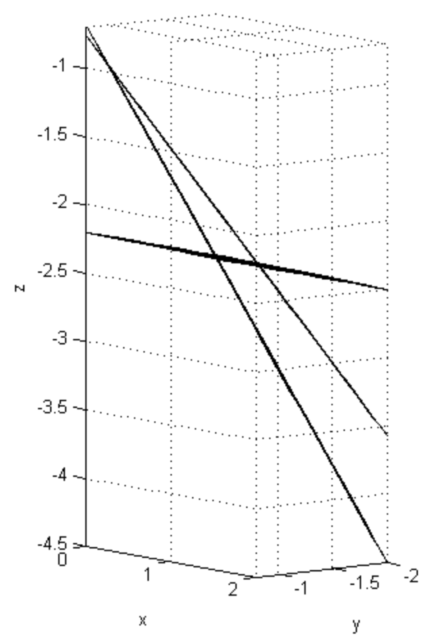
$$\mathbf{n}_1 = \langle 0, 1, -4 \rangle, \quad \mathbf{n}_2 = \langle 2, -3, 2 \rangle, \quad \mathbf{n}_3 = \langle 5, -8, 7 \rangle.$$

Since the normal vectors are not parallel, the planes are not parallel. Getting one point from each plane:

$$P_1 = (0, 8, 0), \quad P_2 = (0, 0, \tfrac{1}{2}), \quad P_3 = (0, 0, \tfrac{1}{7}).$$

With the normal vectors and points we can use the `PlotPlaneFunction`. Below are the graphs of the planes. The second graph looks down the middle of the triangular cross-section of the first graph.





Detailed Solutions to Selected Problems

Chapter 1.1

2 (a)
$$\begin{aligned} 2x + 4y + 6z &= 18 \\ 4x - 6z &= 24 \\ y - 2z &= 4 \end{aligned}$$

3 (a)
$$\left[\begin{array}{cccc|c} 1 & -1 & 2 & -1 & 22 \\ 0 & 3 & -1 & -4 & -2 \\ 1 & -2 & -3 & 0 & 7 \end{array}\right]$$

4 (a)
$$\begin{aligned} x_1 + 3x_2 - x_3 &= 1 \\ x_2 - x_3 &= 1 \\ 2x_1 - x_2 + 5x_3 &= 2 \end{aligned} \rightarrow \left[\begin{array}{ccc|c} 1 & 3 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 2 & -1 & 5 & 2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & -7 & 7 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 7 \end{array}\right]$$

The last row represents the equation $0x_1 + 0x_2 + 0x_3 = 7$, which has no solutions. The system has no solutions and is therefore **inconsistent**.

(b)
$$\begin{aligned} x_1 + 3x_2 - x_3 &= 1 \\ x_2 - x_3 &= 1 \\ 2x_1 - x_2 + 5x_3 &= -5 \end{aligned} \rightarrow \left[\begin{array}{ccc|c} 1 & 3 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 2 & -1 & 5 & -5 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & -7 & 7 & -7 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

The last row represents the equation $0x_1 + 0x_2 + 0x_3 = 0$, which has infinitely many solutions. Let x_3 be free. The second row represents the equation $x_2 - x_3 = 1$ so $x_2 = x_3 + 1$. The first row represents the equation $x_1 + 3x_2 - x_3 = 1 \rightarrow x_1 + 3(x_3 + 1) - x_3 = 1 \rightarrow x_1 + 2x_3 + 3 = 1 \rightarrow x_1 = -2x_3 - 2$. So, letting the free variable $x_3 = t$, the **general solution looks like** $(x_1, x_2, x_3) = (-2t - 2, t + 1, t)$ **for** $-\infty < t < \infty$. One such solution can be obtained by letting $t = 0$, in which case a **particular solution** is $(-2, 1, 0)$, but there are infinitely many.

(c)
$$\begin{aligned} x_1 - 3x_3 &= -5 \\ 2x_1 + x_2 + 2x_3 &= 7 \\ 3x_1 + 2x_2 + x_3 &= 7 \end{aligned}$$

$$\to \left[\begin{array}{ccc|c} 1 & 0 & -3 & -5 \\ 2 & 1 & 2 & 7 \\ 3 & 2 & 1 & 7 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & -5 \\ 0 & 1 & 8 & 17 \\ 0 & 2 & 10 & 22 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & -5 \\ 0 & 1 & 8 & 17 \\ 0 & 0 & -6 & -12 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -3 & -5 \\ 0 & 1 & 8 & 17 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

The third row represents the equation $x_3 = 2$. The second row represents the equation $x_2 + 8x_3 = 17 \to x_2 + 8(2) = 17 \to x_2 = 1$. The first equation represents the equation $x_1 - 3x_3 = -5 \to x_1 - 3(2) = -5 \to x_1 = 1$. Because there is a solution, the system is consistent. The system has the **unique solution** $(x_1, x_2, x_3) = (1, 1, 2)$.

6 (a) $$\begin{array}{rcl} x + 2y + 3z &=& 6 \\ 2y - z &=& 1 \\ x + 4y + 2z &=& 7 \end{array} \to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & 2 & -1 & 1 \\ 1 & 4 & 2 & 7 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 2 & -1 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

The last row represents the equation $0x + 0y + 0z = 0$, which has infinitely many solutions.
If you let $z = t$ be a free parameter, the second row represents $2y - z = 1$ or $y = \frac{1}{2} + \frac{t}{2}$. Subbing both of these into equation 1: $x + 2y + 3z = 6 \to x + (1 + t) + 3t = 6 \to x = 5 - 4t$. So, the **general solution** is $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 - 4t \\ \frac{1}{2} + \frac{t}{2} \\ t \end{bmatrix}$. Letting $t = 0$, a **specific solution** is $\begin{bmatrix} 5 \\ 0.5 \\ 0 \end{bmatrix}$.

If you let $y = t$ be a free parameter, the second row represents $2y - z = 1$ or $z = 2t - 1$. Subbing both these into equation 1: $x + 2y + 3z = 6 \to x + 2t + 3(2t - 1) = 6 \to x = 9 - 8t$. So, the **general solution** is $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9 - 8t \\ t \\ 2t - 1 \end{bmatrix}$. Letting $t = 0$, a **specific solution** is $\begin{bmatrix} 9 \\ 0 \\ -1 \end{bmatrix}$.

If you can use MATLAB® 's *least squares solution* for a **specific solution** $= [0, 1.125, 1.25]^T$.

(b) $$\begin{array}{rcl} x + 2y + 3z &=& 2 \\ 2y - z &=& -3 \\ x + 4y + 2z &=& -5 \end{array} \to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & 2 & -1 & -3 \\ 1 & 4 & 2 & -5 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & 2 & -1 & -3 \\ 0 & 2 & -1 & -7 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & 2 & -1 & -3 \\ 0 & 0 & 0 & -4 \end{array}\right]$$

The last row represents the equation $0x + 0y + 0z = -4$. **No Solutions!**

(c) $$\begin{array}{rcl} x + 2y + 3z &=& -8 \\ 2y - z &=& 10 \\ x + 4y + z &=& 6 \end{array}$$

$$\to \left[\begin{array}{ccc|c} 1 & 2 & 3 & -8 \\ 0 & 2 & -1 & 10 \\ 1 & 4 & 1 & 6 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & -8 \\ 0 & 2 & -1 & 10 \\ 0 & 2 & -2 & 14 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & -8 \\ 0 & 2 & -1 & 10 \\ 0 & 0 & -1 & 4 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & -8 \\ 0 & 2 & -1 & 10 \\ 0 & 0 & 1 & -4 \end{array}\right]$$

So, the last row says $z = -4$, the second row says $2y - z = 10 \to y = 3$, and the first row says $x + 2y + 3z = -8 \to x + 6 - 12 = -8 \to x = -2$. So, $(x, y, z) = (-2, 3, -4)$.

Chapter 1.2

1a $A^T = \begin{bmatrix} 2 & 0 \\ -5 & 4 \end{bmatrix} \qquad B^T = \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix}$

1b $1/2A = \frac{1}{2}\begin{bmatrix} 2 & -5 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & -2.5 \\ 0 & 2 \end{bmatrix}$

1c $A + B = \begin{bmatrix} 2+3 & -5+2 \\ 0-1 & 4+1 \end{bmatrix} = \begin{bmatrix} 5 & -3 \\ -1 & 5 \end{bmatrix}$

1d $2(A+B) = 2\begin{bmatrix} 5 & -3 \\ -1 & 5 \end{bmatrix} = \begin{bmatrix} 10 & -6 \\ -2 & 10 \end{bmatrix}$

$$2A + 2B = 2\begin{bmatrix} 2 & -5 \\ 0 & 4 \end{bmatrix} + 2\begin{bmatrix} 3 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & -10 \\ 0 & 8 \end{bmatrix} + \begin{bmatrix} 6 & 4 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} 10 & -6 \\ -2 & 10 \end{bmatrix}$$

Notice, as the distributive property states: $2(A+B) = 2A + 2B$.

1e $AB = \begin{bmatrix} 2(3)-5(-1) & 2(2)-5(1) \\ 0(3)+4(-1) & 0(2)+4(1) \end{bmatrix} = \begin{bmatrix} 11 & -1 \\ -4 & 4 \end{bmatrix}$

$$BA = \begin{bmatrix} 3(2)+2(0) & 3(-5)+2(4) \\ -1(2)+1(0) & -1(-5)+1(4) \end{bmatrix} = \begin{bmatrix} 6 & -7 \\ -2 & 9 \end{bmatrix}$$

Notice $AB \neq BA$.

1f $Ax = \begin{bmatrix} 2 & -5 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 2(2)-5(-3) \\ 0(2)+4(-3) \end{bmatrix} = \begin{bmatrix} 19 \\ -12 \end{bmatrix}$

$$Bx = \begin{bmatrix} 3 & 2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 3(2)+2(-3) \\ -1(2)+1(-3) \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$$

3a $A^T = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 4 & 0 \\ 0 & -2 & 3 \end{bmatrix}, \quad C^T = \begin{bmatrix} 3 & 1 & -5 \\ 1 & 2 & 0 \end{bmatrix}$

3b $AB = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 4 & -2 \\ 1 & 0 & 3 \end{bmatrix}\begin{bmatrix} 3 & 2 & -1 \\ 0 & 1 & 0 \\ -1 & 2 & 3 \end{bmatrix} =$

$$\begin{bmatrix} 1(3)+2(0)+0(-1) & 1(2)+2(1)+0(2) & 1(-1)+2(0)+0(3) \\ 0(3)+4(0)-2(-1) & 0(2)+4(1)-2(2) & 0(-1)+4(0)-2(3) \\ 1(3)+0(0)+3(-1) & 1(2)+0(1)+3(2) & 1(-1)+0(0)+3(3) \end{bmatrix} = \begin{bmatrix} 3 & 4 & -1 \\ 2 & 0 & -6 \\ 0 & 8 & 8 \end{bmatrix}$$

$$BA = \begin{bmatrix} 3 & 2 & -1 \\ 0 & 1 & 0 \\ -1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 4 & -2 \\ 1 & 0 & 3 \end{bmatrix} =$$

$$\begin{bmatrix} 3(1)+2(0)-1(1) & 3(2)+2(4)-1(0) & 3(0)+2(-2)-1(3) \\ 0(1)+1(0)+0(1) & 0(2)+1(4)+0(0) & 0(0)+1(-2)+0(3) \\ -1(1)+2(0)+3(1) & -1(2)+2(4)+3(0) & -1(0)+2(-2)+3(3) \end{bmatrix} = \begin{bmatrix} 2 & 14 & -7 \\ 0 & 4 & -2 \\ 2 & 6 & 5 \end{bmatrix}$$

Notice: $AB \neq BA$.

3c A is 3x3 and C is 3x2 so AC is the 3x2 matrix given by

$$AC = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 4 & -2 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ -5 & 0 \end{bmatrix} =$$

$$\begin{bmatrix} 1(3)+2(1)+0(-5) & 1(1)+2(2)+0(0) \\ 0(3)+4(1)-2(-5) & 0(1)+4(2)-2(0) \\ 1(3)+0(1)+3(-5) & 1(1)+0(2)+3(0) \end{bmatrix} = \begin{bmatrix} 5 & 5 \\ 14 & 8 \\ -12 & 1 \end{bmatrix}$$

C is a 3x2 and A is a 3x3 so CA is **undefined.**

3d $$Ax = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 4 & -2 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ -3 \\ 0 \end{bmatrix} = \begin{bmatrix} 1(2)+2(-3)+0(0) \\ 0(2)+4(-3)-2(0) \\ 1(2)+0(-3)+3(0) \end{bmatrix} = \begin{bmatrix} -4 \\ -12 \\ 2 \end{bmatrix}$$

C is 3x2 and x is 3x1 so Cx is **undefined**.

3e $x \cdot y = (2)(1) + (-3)(2) + (0)(-1) = \mathbf{-4}$

3f $$3 \begin{bmatrix} 2 \\ -3 \\ 0 \end{bmatrix} - 2 \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 6 \\ -9 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ -4 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -13 \\ 2 \end{bmatrix}$$

5a $$\begin{aligned} 5x_1 + 7x_2 + x_3 + x_4 + x_5 &= 5 \\ -2x_2 + 4x_3 + 2x_4 &= -1 \\ 4x_1 - 2x_3 + 3x_5 &= 3 \end{aligned}$$

6 (a) $$A = \begin{bmatrix} 1 & 0 & -3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix} \quad x = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad b = \begin{bmatrix} -5 \\ 7 \\ 7 \end{bmatrix}$$

7 (a) $$\begin{aligned} x - 3z &= -5 \\ 2x + y + 2z &= 7 \\ 3x + 2y + z &= 7 \end{aligned} \qquad \text{Let } A = \begin{bmatrix} 1 & 0 & -3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} -5 \\ 7 \\ 7 \end{bmatrix}.$$

Then $\mathbf{x} = A \setminus \mathbf{b} = [1, 1, 2]^T$.

And the solution is $\boxed{x = 1,\ y = 1, \text{ and } z = 2}$.

(b) $$\begin{aligned} x_1 + x_2 + 2x_3 &= 8 \\ -x_1 - 2x_2 + 3x_3 &= 1 \\ 3x_1 - 7x_2 + 4x_3 &= 10 \end{aligned} \qquad \text{Let } A = \begin{bmatrix} 1 & 1 & 2 \\ -1 & -2 & 3 \\ 3 & -7 & 4 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} 8 \\ 1 \\ 10 \end{bmatrix}.$$

Then $\mathbf{x} = A \setminus \mathbf{b} = [3, 1, 2]^T$.
And the solution is $\boxed{x_1 = 3,\ x_2 = 1, \text{ and } x_3 = 2}$.

Chapter 1.3

1a Determinant: If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, then $\det(A) = ad - bc$.

So, if $A = \begin{bmatrix} 1 & 2 \\ 4 & 7 \end{bmatrix}$, then $\det(A) = (1)(7) - (2)(4) = -\mathbf{1}$

Inverse: If $\quad A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad$ then $\quad A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

So, if $A = \begin{bmatrix} 1 & 2 \\ 4 & 7 \end{bmatrix} \quad$ then $\quad A^{-1} = \frac{1}{-1} \begin{bmatrix} 7 & -2 \\ -4 & 1 \end{bmatrix} = \begin{bmatrix} -7 & 2 \\ 4 & -1 \end{bmatrix}$

1c Determinant: Performing a cofactor expansion of C along the first row we get

$$\begin{vmatrix} 1 & 0 & -2 \\ -3 & 1 & 4 \\ 2 & -3 & 4 \end{vmatrix} = (-1)^{(1+1)}(1) \begin{vmatrix} 1 & 4 \\ -3 & 4 \end{vmatrix} + (-1)^{(1+2)}(0) \begin{vmatrix} \text{who} \\ \text{cares} \end{vmatrix} + (-1)^{(1+3)}(-2) \begin{vmatrix} -3 & 1 \\ 2 & -3 \end{vmatrix}$$

$= (1)(4 + 12) - 2(9 - 2) = 16 - 14 = \mathbf{2}$.

Inverse: Setting up the augmented matrix:

$$[C|I_3] = \left[\begin{array}{ccc|ccc} 1 & 0 & -2 & 1 & 0 & 0 \\ -3 & 1 & 4 & 0 & 1 & 0 \\ 2 & -3 & 4 & 0 & 0 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|ccc} 1 & 0 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 3 & 1 & 0 \\ 0 & -3 & 8 & -2 & 0 & 1 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 0 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 3 & 1 & 0 \\ 0 & 0 & 2 & 7 & 3 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|ccc} 1 & 0 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 3 & 1 & 0 \\ 0 & 0 & 1 & \frac{7}{2} & \frac{3}{2} & \frac{1}{2} \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 8 & 3 & 1 \\ 0 & 1 & 0 & 10 & 4 & 1 \\ 0 & 0 & 1 & \frac{7}{2} & \frac{3}{2} & \frac{1}{2} \end{array}\right] = [I_3|C^{-1}] \quad \text{so} \quad C^{-1} = \begin{bmatrix} 8 & 3 & 1 \\ 10 & 4 & 1 \\ \frac{7}{2} & \frac{3}{2} & \frac{1}{2} \end{bmatrix}$$

2b This matrix does not have an inverse if the determinant is zero. So we must find the value of a that makes $\det(A) = 0$. Well, $\det(A) = (1 - a)(4 - a) + 2$. Setting this equal to zero and solving for a you get

$$\begin{aligned} (1-a)(4-a) + 2 &= 0 \\ (4 - 5a + a^2) + 2 &= 0 \\ a^2 - 5a + 6 &= 0 \\ (a-2)(a-3) &= 0 \end{aligned}$$

So the two solutions are $a = 2$ and $a = 3$.

2d This matrix does not have an inverse if the determinant is zero. So we must find the value of a that makes $\det(A) = 0$. Performing a cofactor expansion of C along the first row we get

$$\begin{vmatrix} 1 & a & 2 \\ 4 & -1 & 3 \\ -2 & 4 & -1 \end{vmatrix} = (-1)^{(1+1)}(1)\begin{vmatrix} -1 & 3 \\ 4 & -1 \end{vmatrix} + (-1)^{(1+2)}(a)\begin{vmatrix} 4 & 3 \\ -2 & -1 \end{vmatrix} + (-1)^{(1+3)}(2)\begin{vmatrix} 4 & -1 \\ -2 & 4 \end{vmatrix}$$

$= (1)(1-12) - (a)(-4+6) + (2)(16-2) = -11 - 2a + 28 = 17 - 2a.$

Setting this equal to zero yields $a = 17/2 = \mathbf{8.5}$.

3a Let `A = [0, 1, 2; 1, 0, 3; 4, -3, 8]`,
then `b1 = [3, 1, 0]'`, `b2 = [1,1,1]'`, and `b3 = [0,0,0]'`.
Then solving each equation by `inv(A)*b1, inv(A)*b2`, and `inv(A)*b3` you get:

$$x_1 = \begin{bmatrix} -6.5 \\ -2 \\ 2.5 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

3b Let `A = [0, 1, 2; 1, 0, 3; 4, -3, 8]`,
then `x1 = A\b1, x2 = A\b2, x3 = A\b3` you get the same solutions found in part (a).

3c Using MATLAB® to calculate `inv(A)*B` yields
$X = \begin{bmatrix} -6.5 & 1 & 0 \\ -2 & 1 & 0 \\ 2.5 & 0 & 0 \end{bmatrix}$ and the columns of X are the solutions found in parts (a) and (b).

4 Any square matrices (with inverses) can be used to demonstrate the following inverse theorems in MATLAB® :

- `inv(A*B) = inv(B) * inv(A)`
- `inv(inv(A)) = A`
- `inv(A') = (inv(A))'`
- `inv(k*A) = 1/k * inv(A)`

Any square matrices can be used to demonstrate the following determinant theorems in MATLAB® :

- If A is a triangular matrix (upper or lower) then `det(A) = A(1,1) * A(2,2) ... * A(n,n)`.
- `det(A') = det(A)`
- `det(A*B) = det(A) * det(B)`
- `det(inv(A)) = 1/det(A)` provided `det(A)` $\neq$ `0`

5a

$$\begin{aligned} Ax + y &= b \\ Ax &= b - y \\ A^{-1}Ax &= A^{-1}(b - y) \\ I_3 x &= A^{-1}(b - y) \\ x &= A^{-1}(b - y) \\ x &= [-4, 0, 2]^T \end{aligned}$$

5c

$$\begin{aligned} A(Bx + y) &= b \\ Bx + y &= A^{-1}b \\ Bx &= A^{-1}b - y \\ x &= B^{-1}(A^{-1}b - y) \\ x &= [-15, 13, -2]^T. \end{aligned}$$

5e

$$\begin{aligned} ABx &= 2Ax + b \\ ABx - 2Ax &= b \\ (AB - 2A)x &= b \\ x &= (AB - 2A)^{-1}b \\ x &= \text{no solutions.} \end{aligned}$$

Here, det(AB - 2A) = 0 and so there is not a unique solution. In this case there is no solution.

5g

$$\begin{aligned} ABx &= BAy \\ x &= (AB)^{-1}BAy \\ x &= [83.5, 721.5, 308.5]^T. \end{aligned}$$

Notice $(AB)^{-1}BA \neq I_3$.

Chapter 1.4

1 The left side of the equals sign is

$$a_1 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} + a_2 \begin{bmatrix} -2 \\ 1 \\ -3 \end{bmatrix} + a_3 \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 \\ 0 \\ 2a_1 \end{bmatrix} + \begin{bmatrix} -2a_2 \\ a_2 \\ -3a_2 \end{bmatrix} + \begin{bmatrix} 2a_3 \\ 5a_3 \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 - 2a_2 + 2a_3 \\ a_2 + 5a_3 \\ 2a_1 - 3a_2 \end{bmatrix}$$

The right side of the equals sign is

$$\begin{bmatrix} 1 & -2 & 2 \\ 0 & 1 & 5 \\ 2 & -3 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_1 - 2a_2 + 2a_3 \\ a_2 + 5a_3 \\ 2a_1 - 3a_2 \end{bmatrix}$$

These are indeed equal.

2a Here we must solve the system $Ma = x$ where $M = \begin{bmatrix} 1 & 2 \\ 4 & 7 \end{bmatrix}$, $a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$, and $x = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$.

Using the formula for the inverse of a $2x2$ matrix or letting MATLAB® do it. You get

$$M^{-1} = \begin{bmatrix} -7 & 2 \\ 4 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = M^{-1}x = \begin{bmatrix} -7 & 2 \\ 4 & -1 \end{bmatrix} \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 20 \\ -11 \end{bmatrix}$$

Or, you can use the MATLAB® command `a = M\x` and get the same answer.

Either way, $a_1 = 20$**, and** $a_2 = -11$.

3a Here we must solve the system $Ma = x$ where

$$M = \begin{bmatrix} 1 & 0 & -2 \\ -3 & 1 & 4 \\ 2 & -3 & 4 \end{bmatrix}, \; a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \text{ and } x = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Calculating the inverse of M or letting MATLAB® do you it. You get

$$M^{-1} = \begin{bmatrix} 8 & 3 & 1 \\ 10 & 4 & 1 \\ 3.5 & 1.5 & 0.5 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = M^{-1}x = \begin{bmatrix} 8 & 3 & 1 \\ 10 & 4 & 1 \\ 3.5 & 1.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 12 \\ 15 \\ 5.5 \end{bmatrix}$$

Or, you can use the MATLAB® command `a = M\x` and get the same answer.

Either way, $a_1 = 12$**,** $a_2 = 15$**, and** $a_3 = 5.5$.

Chapter 1.5

1a Create the matrix M:

$$M = \left[\begin{array}{c|c|c} \vdots & \vdots & \vdots \\ v_1 & v_2 & v_3 \\ \vdots & \vdots & \vdots \end{array}\right] = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 5 \\ 3 & 5 & 11 \end{bmatrix} \text{ and } \det(M) = 1(11-25) - 0(\text{who cares}) + 2(10-3) = 0.$$

Here we use the **Super Theorem - Version 2** and conclude that since $\det(M) = 0$, the vectors **are not linearly independent** and **do not form a basis** for $\mathbb{R}^3$.

Now, can we find a nonzero linear combination of these vectors that result in the zero vector? To do this we need to find set up the equation: $Ma = 0$ and solve for a. Putting this equation in augmented matrix form:

$$\left[\begin{array}{ccc|c} 1 & 0 & 2 & 0 \\ 2 & 1 & 5 & 0 \\ 3 & 5 & 11 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 5 & 5 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

.

The last equation is useless and gives us a free variable a_3. The second equation says $a_2 + a_3 = 0$ or $a_2 = -a_3$. The first equations says that $a_1 + 2a_3 = 0$ or $a_1 = -2a_3$. Letting a_3 be the free parameter t we get $[a_1, a_2, a_3] = [-2t, -t, t]$ for $-\infty < t < \infty$. Letting t be any number other than zero can get a non-zero solution. For example, if we let $t = 1$, the solution becomes $a = [a_1, a_2, a_3] = [-2, -1, 1]$. So notice, $-2u_1 - u_2 + u_3 = [0, 0, 0]$ and we have a non-zero linear combination of the vectors which produces the zero vector.

1c We know that that there are not enough vectors to form a basis for $\mathbb{R}^3$. Can we prove that they are linearly independent? We need to determine if there is a non-zero linear combination of these vectors that result in the zero vector. That is we must solve the system:

$$a_1v_1 + a_2v_2 = \mathbf{0} \quad \rightarrow \quad \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Expressing this system in augmented matrix form:

$$\left[\begin{array}{cc|c} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 5 & 0 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 0 \end{array}\right]$$

The last equation states that $5a_2 = 0$, so $a_2 = 0$. The first equation states that $a_1 = 0$. As such, the only solution is $[a_1, a_2] = [0, 0]$ and the vectors are linearly independent.

2a Use these vectors as the columns in the matrix V:

$$V = \begin{bmatrix} 1 & 1 & 3 \\ -1 & 0 & -4 \\ 2 & 3 & 5 \end{bmatrix} \text{ and } \det(V) = 1(0 + 12) - 1(-5 + 8) + 3(-3 - 0) = 0.$$

Since the determinant of V is 0, we use **Super Theorem - Version 2** and conclude that the columns are not linearly independent and do not form a basis for $\mathbb{R}^3$.

Chapter 1.6

1a Let $V = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$ (the matrix with the new basis as column vectors).

Then the transition matrix is $T = V^{-1} = \begin{bmatrix} -3 & 2 & 0 \\ 2 & -1 & 0 \\ 1.5 & -1 & 0.5 \end{bmatrix}$ (using MATLAB®). The new point is

$TP = \begin{bmatrix} -3 & 2 & 0 \\ 2 & -1 & 0 \\ 1.5 & -1 & 0.5 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} -6 \\ 4 \\ 4.5 \end{bmatrix}$. So, with respect to the new basis, $P = (-6, 4, 4.5)$.

2a Consider the radius vector $r = [3, 2]$ given with respect to the standard basis. If we choose a new basis consisting of $v_1 = [3, 0]^T$ and $v_2 = [0, 2]^T$ the radius vector is $1v_1 + 1v_2$. So, with respect to this new basis (called Ellipse Space), the radius vector is simply $r_{new} = [1, 1]$. In other words, our ellipse has been transformed into a unit circle. Now we want to transform the point $p = [1.8, 1.8]$ into Ellipse Space. So, the transition matrix is given by

$$T = V^{-1} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

So we multiply p on the left by T to get p_{new}:

$$p_{new} = Tp = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1.8 \\ 1.8 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.9 \end{bmatrix}$$

So, now we just check to see if p_{new} is within the unit circle. So look at $x_{new}^2 + y_{new}^2 = (0.6)^2 + (0.9)^2 = 1.17$. Since this value is greater than 1, we conclude that p_{new} is outside of the transformed ellipse and so **the point p is outside of the ellipse**.

3a Consider the radius vector $r = [1, 2, 3]$ given with respect to the standard basis. If we choose a new basis consisting of $v_1 = [1, 0, 0]^T$, $v_2 = [0, 2, 0]^T$, and $v_3 = [0, 0, 3]^T$ the radius vector is $1v_1 + 1v_2 + 1v_3$. So, with respect to this new basis (called Ellipsoid Space), the radius vector is simply $r_{new} = [1, 1, 1]$. In other words, our ellipsoid has been transformed into a unit sphere. Now we want to transform the point $p = [0.5, 1.5, 2.0]$ into Ellipsoid Space. So, the transition matrix is given by

$$T = V^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

So we multiply p on the left by T to get p_{new}:

$$p_{new} = Tp = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.5 \\ 1.5 \\ 2.0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.75 \\ 0.67 \end{bmatrix}$$

So, now we just check to see if p_{new} is within the unit sphere. Look at $x_{new}^2 + y_{new}^2 + z_{new}^2 = (0.5)^2 + (0.75)^2 + (0.67)^2 \approx 1.26$. Since this value is greater than 1, we conclude that p_{new} is outside the unit sphere and so **the original point p is outside of the ellipsoid**.

4 The original line has slope $= \frac{5}{7}$ and y-intercept $= 4$. The transition matrix is given by

$$T = V^{-1} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

Now, let's translate a couple points on the line, say $p = [0, 4]$ and $q = [7, 9]$. Translating these into *Ellipse Space:*

$$p_{new} = Tp = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad \text{and} \quad q_{new} = Tq = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 7 \\ 9 \end{bmatrix} = \begin{bmatrix} 7/3 \\ 9/2 \end{bmatrix}$$

So the transformed points in ellipse space are (0,2) and (7/3, 9/2). Since we have two points, we can get the equation for this line as $y_{new} = m_{new}x_{new} + b_{new}$ where

$$m_{new} = \frac{9/2-2}{7/3-0} = \frac{5/2}{7/3} = \frac{5\cdot\frac{1}{2}}{7\cdot\frac{1}{3}}\cdot = \frac{15}{14}$$

.

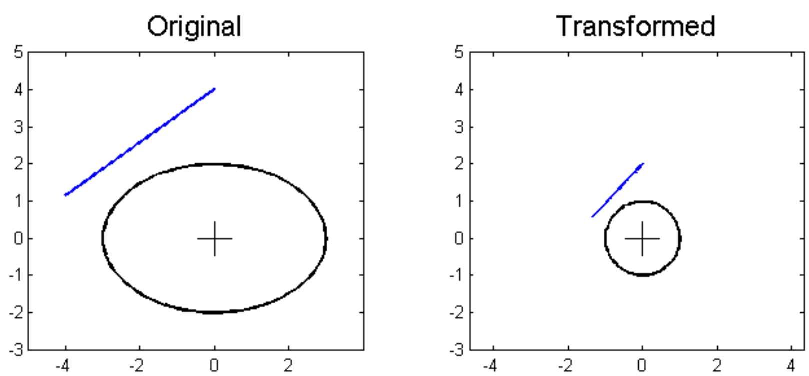
The original slope was $\frac{5}{7}$. After the transformation, ΔY is changed by a factor of $\frac{1}{2}$ and Δx is changed by a factor of $\frac{1}{3}$ as the transition matrix would suggest.

The new y-intercept is given by p_{new} which is 2. Notice the original y-intercept is changed by a factor of $\frac{1}{2}$ as the transition matrix would suggest.

The equation of the transformed line with respect to the new basis is

$$y_{new} = \frac{15}{14}\,x_{new} + 2.$$

The slope and intercept are modified in the transformation according the the transition matrix T.



Chapter 1.7

1a The original point is denoted in vector form as $[0,3]^T$. First, you have to add 1 unit to the x-value and 2 units to the y-value. Second, The rotation matrix is $R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ which rotates points in the clock-wise direction by θ radians. In this case, $\theta = -60^o$ which is $-60\frac{\pi}{180} = -\frac{\pi}{3}$ radians. The translation occurs first so the complete transformation is

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} 0 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} \approx \begin{bmatrix} -3.8 \\ 3.4 \end{bmatrix}$$

Which must be done in MATLAB® .

2a The original point is denoted in vector form as $[-1,3]^T$ and $\theta = \pi/6$.
The rotation matrix: $Rot = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$. The reflection matrix: $Ref = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

The rotation is done first:

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = [Ref]\,[Rot] \begin{bmatrix} -1 \\ 3 \end{bmatrix} \approx \begin{bmatrix} 0.6 \\ -3.1 \end{bmatrix}$$

Which must be done in MATLAB® .

3a The translation vector is $[-2, 3]^T$ for 2 units left ($\Delta x = -2$) and 3 units up ($\Delta y = 3$).
The rotation matrix: $Rot = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$
where $\theta = -45^o = -\pi/4$. The angle (θ) is negative because the rotation is counter-clockwise.
The rotation occurs first so the transformation looks like

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

You must solve this for $[x, y]^T$.

$$\begin{aligned} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix} &= \begin{bmatrix} 0 \\ 5 \end{bmatrix} \\ \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 0 \\ 5 \end{bmatrix} - \begin{bmatrix} -2 \\ 3 \end{bmatrix} \\ \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 2 \\ 2 \end{bmatrix} \end{aligned}$$

This must be solved with MATLAB® (using the inverse or the \ command). You should get $[x, y]^T = [2.8,\ 0.0]$. So, $(x, y) = (\mathbf{2.8}, \mathbf{0.0})$.

4a The rotation occurs first so the transformation looks like

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$$

So, we must solve the equation above for $[x, y]^T$. There are many ways to do this but all of them require MATLAB® (using inverses or the \ command). You should get $[x, y]^T = [-0.23,\ -3.60]$. So, $(x, y) = (\mathbf{-0.23}, \mathbf{-3.60})$.

5 (a) $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(-\pi/4) & \sin(-\pi/4) \\ -\sin(-\pi/4) & \cos(-\pi/4) \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix} \approx \begin{bmatrix} 0.71 \\ -2.12 \end{bmatrix}$$

(b) $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(-\pi/4) & \sin(-\pi/4) \\ -\sin(-\pi/4) & \cos(-\pi/4) \end{bmatrix} \left(\begin{bmatrix} -1 \\ -2 \end{bmatrix} - \begin{bmatrix} 0 \\ 3 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 3 \end{bmatrix} \approx \begin{bmatrix} 2.83 \\ -1.24 \end{bmatrix}$$

Chapter 2.1

1a $\mathbf{v} = \langle -3, -10\rangle$

2a $\mathbf{u} = \langle -2, -5\rangle$ and $\mathbf{v} = \langle -4, -1\rangle$, not equivalent, not parallel.

2c $\mathbf{u} = \langle 3, -2\rangle$ and $\mathbf{v} = \langle -6, 4\rangle$, not equivalent but $\mathbf{v} = -2\ \mathbf{u}$ therefore parallel.

3a $\overrightarrow{PQ} = \langle 3, 6\rangle$ and $\overrightarrow{PR} = \langle 2, 4\rangle$. Notice, $\overrightarrow{PR} = \frac{2}{3}\overrightarrow{PQ}$ so these vectors are parallel and the points are collinear.

5a $\overrightarrow{PQ} = \langle 3, 4\rangle$ and $\overrightarrow{PR} = \langle 8, w-3\rangle$. If $\overrightarrow{PR} = k\ \overrightarrow{PQ}$ then $3\ k = 8$ and $4\ k = w - 3$. Solving the first for k you get $k = 8/3$,then plugging this into the second you get $4\ \frac{8}{3} = w - 3$. Solving this for w you get $w = \frac{32}{3} + 3 = \frac{41}{3}$.

A faster way: We need $\dfrac{8}{3} = \dfrac{w-3}{4}$. You get the same answer.

6a $= \langle 0, -1\rangle$

6b $= \langle 4, 7\rangle$ Note: $\mathbf{v} - \mathbf{u} = \langle -4, -7\rangle = -(\mathbf{u} - \mathbf{v})$.

6c $||\mathbf{u}|| = \sqrt{2^2 + 3^2} = \sqrt{13}$ and $||\mathbf{v}|| = \sqrt{(-2)^2 + (-4)^2} = \sqrt{20} = 2\sqrt{5}$.

6d $= \sqrt{0^2 + (-1)^2} = 1$.

6e It is true that $1 \le \sqrt{13} + 2\sqrt{5}$.

6f $= \langle -4, -6\rangle + \langle -6, -12\rangle = \langle -10, -18\rangle$

7a $\mathbf{u} = \frac{1}{||\mathbf{v}||}\mathbf{v} = \frac{1}{5}\mathbf{v} = \left\langle \frac{3}{5}, -\frac{4}{5}\right\rangle$ Note: $||\mathbf{u}|| = 1$ as it should.

9 From the previous problem, the unit vector in the direction of $\overrightarrow{PQ} = \mathbf{u} = \left\langle \frac{3}{\sqrt{73}}, \frac{8}{\sqrt{73}}\right\rangle$. So the vector of length 2 in this same direction is $2\mathbf{u} = \left\langle \frac{6}{\sqrt{73}}, \frac{16}{\sqrt{73}}\right\rangle \approx \langle 0.70, 1.87\rangle$.

11a $= 2\ \mathbf{i} + 3\ \mathbf{j}$

12a Since the vector must have length 3, we know that $||\mathbf{u}|| = 3$, and $-120^o = -\frac{2\pi}{3}$.
Expressed as a linear combination of $\mathbf{i}$ and $\mathbf{j}$

$$\begin{aligned}
\mathbf{u} &= ||\mathbf{u}||\ \cos\theta\ \mathbf{i} + ||\mathbf{u}||\ \sin\theta\ \mathbf{j} \\
&= 3\ \cos(-\frac{2\pi}{3})\ \mathbf{i} + 3\ \sin(-\frac{2\pi}{3})\mathbf{j} \\
&= 3\ \frac{-1}{2}\ \mathbf{i} + 3\ \frac{-\sqrt{3}}{2}\ \mathbf{j} \\
&= \frac{-3}{2}\mathbf{i} - \frac{3\sqrt{3}}{2}\ \mathbf{j}
\end{aligned}$$

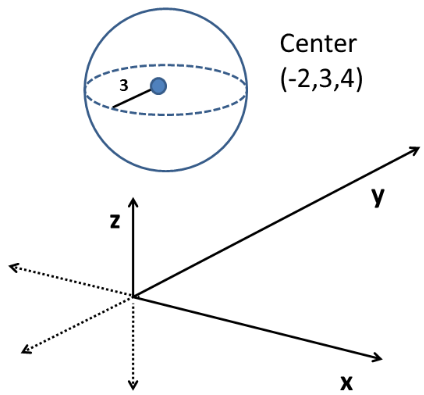
Expressed in component form $\mathbf{u} = \left\langle \frac{-3}{2}, \frac{-3\sqrt{3}}{2}\right\rangle \approx \langle -1.50, -2.60\rangle$.

Chapter 2.2

1a $x^2 + (y+4)^2 + (z-7)^2 = 16$

1c If this sphere is to be tangent to the xz-plane, and the center is 3 units from the xz-plane then the radius must be 3.

$(x+2)^2 + (y-3)^2 + (z-4)^2 = 9$

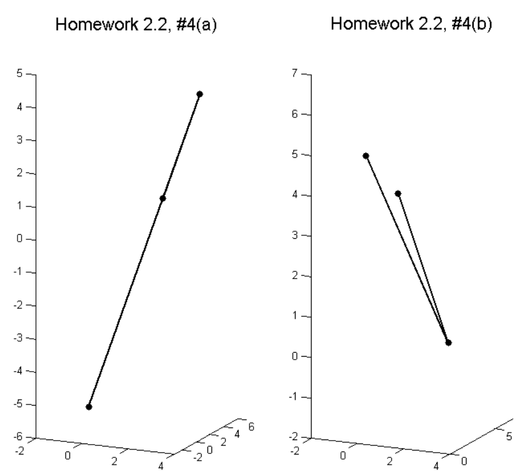


2a $\mathbf{v} = \langle -3, -10, 10 \rangle$

3a $\mathbf{u}\,\langle -2, -5, 2 \rangle$ and $\mathbf{v} = \langle -4, -1, 0 \rangle$, not equivalent, not parallel.

3c $\mathbf{u} = \langle 3, -2, 3 \rangle$ and $\mathbf{v} = \langle -6, 4, -6 \rangle$. Notice $\mathbf{v} = -2\mathbf{u}$ so the vectors are parallel but not equivalent.

4a $\overrightarrow{PQ} = \langle 3, 6, 9 \rangle$ and $\overrightarrow{PR} = \langle 2, 4, 6 \rangle$.
Notice: $\overrightarrow{PR} = \frac{2}{3}\overrightarrow{PQ}$ making the vectors parallel and the **points are collinear**.



5a $\overrightarrow{PQ} = \langle 3, 4, -4\rangle$ and $\overrightarrow{PR} = \langle x+1, y-3, 3\rangle$. In order for these to be parallel we need

$$\frac{x+1}{3} = \frac{y-3}{4} = \frac{-3}{4}$$

Solving $\dfrac{x+1}{3} = \dfrac{-3}{4} \to x+1 = \dfrac{-9}{4} \to x = \dfrac{-9}{4} - 1 = \dfrac{-13}{4}$

Solving $\dfrac{y-3}{4} = \dfrac{-3}{4} \to y-3 = -3 \to y = 0$

So the answers are $\mathbf{x} = \frac{\mathbf{-13}}{\mathbf{4}}$ and $\mathbf{y} = \mathbf{0}$.

6a $= \langle 0, -1, -5\rangle$

6b $= \langle 4, 7, -5\rangle$

6c $||\mathbf{u}|| = \sqrt{2^2 + 3^2 + (-5)^2} = \sqrt{38}$.

$||\mathbf{v}|| = \sqrt{(-2)^2 + (-4)^2 + 0^2} = \sqrt{20} = 2\sqrt{5}$.

6d $= ||\,\langle 0, -1, -5\rangle\,|| = \sqrt{0^2 + (-1)^2 + (-5)^2} = \sqrt{26}$.

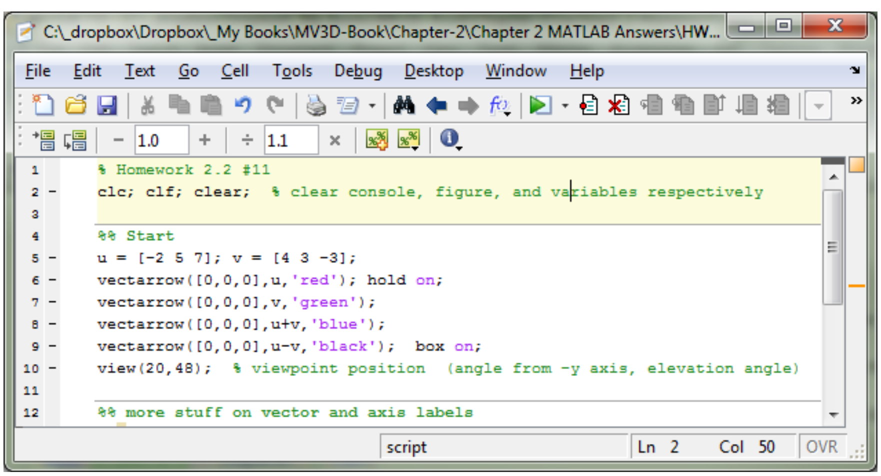
6e It is true that $\sqrt{26} \le 2\sqrt{5} + \sqrt{26}$ or approximately $5.1 \le 9.6$.

6f $= \langle -4, -6, 10\rangle + \langle -6, -12, 0\rangle = \langle -10, -18, 10\rangle$

6g $\dfrac{\mathbf{v}}{||\mathbf{v}||} = \dfrac{1}{\sqrt{20}}\langle -2, -4, 0\rangle = \dfrac{1}{2\sqrt{5}}\langle -2, -4, 0\rangle = \dfrac{1}{\sqrt{5}}\langle -1, -2, 0\rangle$.

8 This is just a follow-up to the last problem, and similar problems in the previous section. We already have the unit vector in the desired direction as $\mathbf{u} = \frac{1}{2\sqrt{38}}\langle 4, 10, 6\rangle$ from the last problem. Let, $\mathbf{v} = 2\mathbf{u}$, and add this to H in vector form $\langle 0, 0, 2\rangle$. So, $\frac{2}{2\sqrt{38}}\langle 4, 10, 6\rangle + \langle 0, 0, 2\rangle \approx \langle 0.65, 1.62, 2.97\rangle$ and the point $T = (0.65, 1.62, 2.97)$.

11 Here I use the plotting function file `vectarrow.m`.



Chapter 2.3

1a $\mathbf{u}\cdot\mathbf{v} = -6 \qquad \mathbf{u}\cdot\mathbf{u} = 25 \qquad ||\mathbf{u}||^2 = 25 \qquad 2(\mathbf{u}\cdot\mathbf{v})\ \mathbf{v} = \langle -24, 36\rangle.$

2a $\mathbf{u}\cdot\mathbf{v} = ||\mathbf{u}||\ ||\mathbf{v}||\ \cos(\theta) = 24\cos(\frac{7}{36}\pi) \approx 19.66.$

3a $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||} = 0$ therefore $\theta = \frac{\pi}{2} = 90^o$

4a $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||} = \frac{4}{4\sqrt{2}} == \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2}.$ Neither, the angle between them is acute because $\cos(\theta) > 0$.

4c $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||} = \frac{-6}{\sqrt{14}\sqrt{3}} = \frac{-6}{\sqrt{42}}.$ Neither, the angle between them is obtuse because $\cos(\theta) < 0$.

5a We need $\mathbf{u}\cdot\mathbf{v} = 0$, and $\mathbf{u}\cdot\mathbf{v} = 3 - 4 + w$. Setting this equal to zero we get $-1 + w = 0$ or $w = 1$.

6a $\text{Proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v} = \left(\frac{4}{2}\right)\langle 1, 1\rangle = \langle 2, 2\rangle$

6c $\text{Proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v} = \left(\frac{-6}{3}\right)\langle -1, 1, -1\rangle = \langle 2, -2, 2\rangle$

8 (a) $\mathbf{u}\cdot(\mathbf{v}+\mathbf{w})$ is well-defined because the sum of two vectors is a vector and then this can be dotted with another vector of the same length.

(b) $\mathbf{u}\cdot\mathbf{v}+\mathbf{w}$ is not defined because $\mathbf{u}\cdot\mathbf{v}$ is a scalar and you can't add a scalar to a vector.

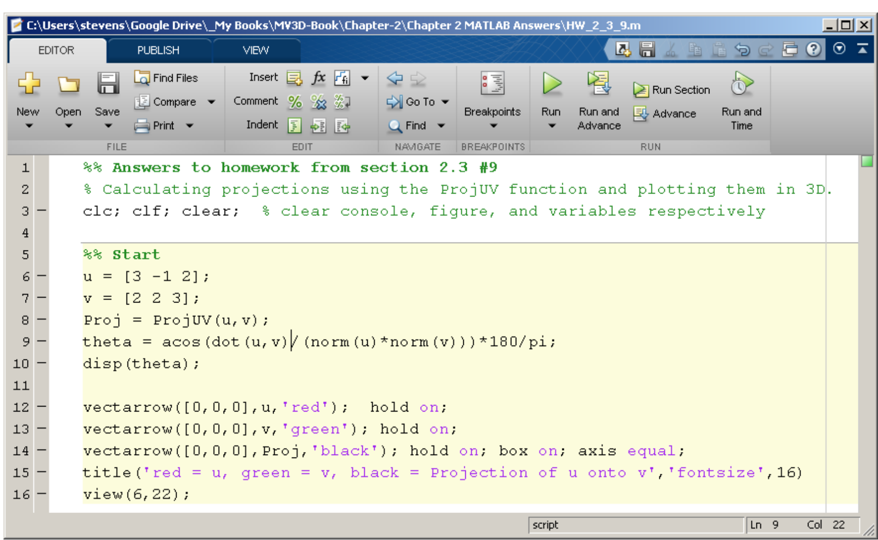
(c) $(\mathbf{u}\cdot\mathbf{v})\ \mathbf{w}$ is well-defined: $(\mathbf{u}\cdot\mathbf{v})$ is a scalar and you can multiply this by the vector $\mathbf{w}$.

(d) $(\mathbf{u}\cdot\mathbf{v})\cdot\mathbf{w}$ is not defined because $(\mathbf{u}\cdot\mathbf{v})$ is a scalar and you can't have a dot product between a scalar and a vector.

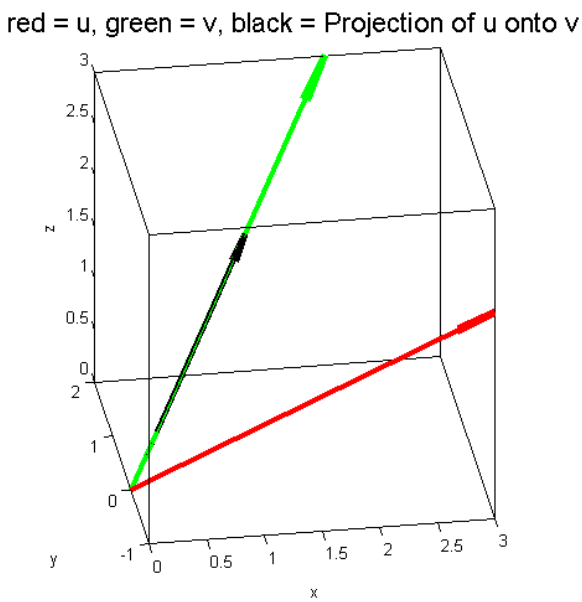
(e) $||\mathbf{u}||(\mathbf{v}+\mathbf{w})$ is well defined: $||\mathbf{u}||$ is a scalar and you can multiply this by the vector $\mathbf{v}+\mathbf{w}$.

(f) $\dfrac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{w}}$ is not defined because you can't divide by a vector.

9 Below is the code and graph for this problem. The angle between the vectors (as solved using the `acos` function is $\theta = 49.6^o$.



```
%% Answers to homework from section 2.3 #9
% Calculating projections using the ProjUV function and plotting them in 3D.
clc; clf; clear;  % clear console, figure, and variables respectively

%% Start
u = [3 -1 2];
v = [2 2 3];
Proj = ProjUV(u,v);
theta = acos(dot(u,v)/(norm(u)*norm(v)))*180/pi;
disp(theta);

vectarrow([0,0,0],u,'red');  hold on;
vectarrow([0,0,0],v,'green'); hold on;
vectarrow([0,0,0],Proj,'black'); hold on; box on; axis equal;
title('red = u, green = v, black = Projection of u onto v','fontsize',16)
view(6,22);
```



10 First you need P at the intersection of $y = x$ and $y = 3x - 6$. Solving these two equations for x and y yields $P(3, 3)$ and the vector $\overrightarrow{PQ} = \langle 2, 2\rangle$. The wall vector $\mathbf{w}$ can be found by choosing any two points on the line $y = 3x - 6$. Using the points (2,0) and (3,3) you get $\mathbf{w} = \langle 1, 3\rangle$.

Slide Vector: Using equation (2.17),

$$\mathbf{s} = \mathrm{Proj}_{\mathbf{w}}\overrightarrow{PQ} = \left(\frac{\mathbf{w}\cdot\overrightarrow{PQ}}{||\mathbf{w}||^2}\right)\mathbf{w} = \frac{\langle 1,3\rangle\cdot\langle 2,2\rangle}{||\,\langle 1,3\rangle\,||^2}\ \langle 1,3\rangle = \frac{8}{10}\ \langle 1,3\rangle = \langle 0.8, 2.4\rangle,$$

and the final slide point $= P + \mathbf{s} = (3,3) + \langle 0.8, 2.4\rangle = \boxed{(3.8,\ 5.4)}$.

Bounce Vector: Using equation (2.18),

$\mathbf{b} = 2\mathbf{s} - \overrightarrow{PQ} = 2\,\langle 0.8, 2.4\rangle - \langle 2, 2\rangle = \langle -0.4, 2.8\rangle$,

and the final bounce point is $= P + \mathbf{b} = (3,3) + \langle -0.4, 2.8\rangle = \boxed{(2.6,\ 5.8)}$.

12a Using, $\mathrm{Proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)\ \mathbf{v}$.

$$\mathrm{Proj}_{(-\mathbf{v})}\mathbf{u} = \left(\frac{\mathbf{u}\cdot(-\mathbf{v})}{||-\mathbf{v}||^2}\right)(-\mathbf{v}) = -\left(\frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{v}||^2}\right)(-\mathbf{v}) = \left(\frac{(\mathbf{u}\cdot\mathbf{v})}{||\mathbf{v}||^2}\right)(\mathbf{v}) = \mathrm{Proj}_{\mathbf{v}}\mathbf{u}$$

Chapter 2.4

1a $\mathbf{v}\times\mathbf{u} = -\mathbf{u}\times\mathbf{v} = -\mathbf{w}$.

1b $-\mathbf{u}\times\mathbf{v} = -(\mathbf{u}\times\mathbf{v}) = -\mathbf{w}$.

1c $||2\ \mathbf{u}\times 2\ \mathbf{v}|| = 4||\mathbf{u}\times\mathbf{v}|| = 4||\mathbf{w}||$.

1d Since $\mathbf{u} \times \mathbf{v}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$ then $\mathbf{w}$ is orthogonal to $\mathbf{u}$ which implies the dot product $\mathbf{w} \cdot \mathbf{u} = 0$.

2a Any multiple of $\mathbf{u} \times \mathbf{v}$ or $\mathbf{v} \times \mathbf{u}$ will be orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.

$$\begin{aligned} \mathbf{u} \times \mathbf{v} &= \langle (u_2v_3 - u_3v_2),\ -(u_1v_3 - u_3v_1),\ (u_1v_2 - u_2v_1) \rangle \\ &= \langle ((-1)(0) - (1)(3)),\ -((1)(0) - (1)(-2)),\ ((1)(3) - (-1)(-2)) \rangle = \langle -3, -2, 1 \rangle \end{aligned}$$

. or, using the matrix determinant method:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & -1 & 1 \\ -2 & 3 & 0 \end{vmatrix} = -3\mathbf{i} - 2\mathbf{j} + 1\mathbf{k} = \langle -3, -2, 1 \rangle .$$

Normalizing this to have length one, a unit vector is $\dfrac{1}{\sqrt{14}} \langle -3, -2, 1 \rangle$. So the two unit vectors orthogonal to both $\mathbf{u}$ and $\mathbf{v}$ are $\dfrac{\pm 1}{\sqrt{14}} \langle -3, -2, 1 \rangle \approx \langle -0.8018,\ -0.5345,\ 0.2673 \rangle$

Chapter 2.5

1a

$$x = 2 + t, \quad y = 3 - 2t, \quad z = 4 + 3t$$

1c If the direction vector is parallel to the xz-plane and the yz-plane then it must be parallel to the z-axis. So a suitable direction vector would be $\langle 0, 0, 1 \rangle$, and the parametric equations for line are

$$x = 2, \quad y = 3, \quad z = 4 + t, \quad \text{for} \quad t \in (-\infty, \infty)$$

2 (a) $\begin{array}{l} x = 4t + 2,\ y = 3,\ z = -t + 1 \\ x = 2s + 2,\ y = 2s + 3,\ z = s + 1 \end{array}$

You need to determine if there are values for t and s where (x, y, z) are the same for both lines. So we need to solve these three equations for the two unknown values of s and t.

$$\begin{array}{rcl} 4t + 2 &=& 2s + 2 \\ 3 &=& 2s + 3 \\ -t + 1 &=& s + 1 \end{array} \quad \sim \quad \begin{array}{rcl} 4t - 2s &=& 0 \\ -2s &=& 0 \\ -t - s &=& 0 \end{array} \quad \sim \quad \begin{bmatrix} 4 & -2 \\ 0 & -2 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Expressing these equations in augmented matrix form, you can use Gaussian Elimination to solve for the variables (or have software do it for you but be careful of the *no solution* problems that can occur).

$$\left[\begin{array}{rr|r} 4 & -2 & 0 \\ 0 & -2 & 0 \\ -1 & -1 & 0 \end{array}\right] \sim \left[\begin{array}{rr|r} -1 & -1 & 0 \\ 0 & -2 & 0 \\ 4 & -2 & 0 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 0 \\ 0 & -2 & 0 \\ 4 & -2 & 0 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & -6 & 0 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & -6 & 0 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

The second equation says that $s = 0$, plugging this into the first equation gives $t = 0$. Notice when $t = 0$ and $s = 0$ in the parametric equations of the lines, they both result in $x = 2$, $y = 3$, and $z = 1$. So the point of intersection is (2,3,1). To find the angle of intersection, you must find the angle between the two direction vectors $\mathbf{u} = \langle 4, 0, -1\rangle$ and $\mathbf{v} = \langle 2, 2, 1\rangle$. You get $\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \frac{7}{\sqrt{17}\,\sqrt{9}} = \frac{7}{3\sqrt{17}}$. Therefore $\theta = \arccos\left(\frac{7}{3\sqrt{17}}\right) \approx 0.97$ radians or 55.5^o.

2 (c) $\begin{array}{l} x = 3t,\ y = -t + 2,\ z = t - 1 \\ x = 4s + 1,\ y = s - 2,\ z = -3s - 3 \end{array}$

You need to determine if there are values for t and s where (x, y, z) are the same for both lines. So we need to solve these three equations for the two unknown values of s and t.

$$\begin{array}{rcl} 3t &=& 4s+1 \\ -t+2 &=& s-2 \\ t-1 &=& -3s-3 \end{array} \quad\sim\quad \begin{array}{rcr} 3t-4s &=& 1 \\ -t-s &=& -4 \\ t+3s &=& -2 \end{array} \quad\sim\quad \begin{bmatrix} 3 & -4 \\ -1 & -1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 1 \\ -4 \\ -2 \end{bmatrix}$$

Expressing these equations in augmented matrix form, you can use Gaussian Elimination to solve for the variables (or have software do it for you but be careful of the *no solution* problems that can occur).

$$\left[\begin{array}{rr|r} 3 & -4 & 1 \\ -1 & -1 & -4 \\ 1 & 3 & -2 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 4 \\ 3 & -4 & 1 \\ 1 & 3 & -2 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 4 \\ 0 & -7 & -11 \\ 0 & 2 & -6 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 4 \\ 0 & 1 & \frac{11}{7} \\ 0 & 2 & -6 \end{array}\right] \sim \left[\begin{array}{rr|r} 1 & 1 & 4 \\ 0 & 1 & \frac{11}{7} \\ 0 & 0 & \frac{-64}{7} \end{array}\right]$$

The last equation says that $0t + 0s = \frac{-64}{7}$. There is no solution to this so the lines do not intersect.

3a We'll use the formula for the distance between a point Q and a line with direction vector $\mathbf{v}$;

$$D = \frac{||\overrightarrow{PQ} \times \mathbf{v}||}{||\mathbf{v}||},$$

where P is any point on the line. The direction vector for our line is $\mathbf{v} = \langle 4, 0, -1\rangle$ and a good choice for P is found by letting $t = 0$ in the parametric equations for the line. Doing this give $P = (-2, 3, 1)$, and therefore $\overrightarrow{PQ} = \langle 3, 2, -3\rangle$. So now in order to use the formula above we must first fine $\overrightarrow{PQ} \times \mathbf{v}$

$$\overrightarrow{PQ} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 3 & 2 & -3 \\ 4 & 0 & -1 \end{vmatrix} = -2\mathbf{i} - 9\mathbf{j} - 8\mathbf{k} = \langle -2, -9, -8\rangle.$$ Now the distance from Q to the line is

$$D = \frac{||\overrightarrow{PQ} \times \mathbf{v}||}{||\mathbf{v}||} = \frac{\sqrt{149}}{\sqrt{17}} \approx 2.96$$

4 (a) $\begin{array}{l} x = 4t + 2,\ y = 6t + 3 \\ x = s + 1,\ y = 2s + 4 \end{array}$

You need to determine if there are values for t and s where (x, y) are the same for both lines. So we need to solve these two equations for the two unknown values of s and t.

$$\begin{array}{rcl} 4t+2 &=& s+1 \\ 6t+3 &=& 2s+4 \end{array} \sim \begin{array}{rcl} 4t-s &=& -1 \\ 6t-2s &=& 1 \end{array} \sim \begin{bmatrix} 4 & -1 \\ 6 & -2 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The determinant is -8 +6 = -2 so there is a unique solution for s and t. You can use Gaussian Elimination or the inverse of the matrix to get the solutions $t = -1.5$ and $s = -5$.

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 4 & -1 \\ 6 & -2 \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \frac{1}{-2} \begin{bmatrix} -2 & 1 \\ -6 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \frac{-1}{2} \begin{bmatrix} 3 \\ 10 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -5 \end{bmatrix}$$

So the point of intersection occurs when $t = -1.5$ or $s = -5$

$x = 4(-1.5) + 2,\ y = 6(-1.5) + 3 \quad \rightarrow \quad (x, y) = (-4, -6)$

$x = -5 + 1,\ y = 2(-5) + 4 \quad \rightarrow \quad (x, y) = (-4, -6)$

To find the angle of intersection, you must find the angle between the two direction vectors $\mathbf{u} = \langle 4, 6 \rangle$ and $\mathbf{v} = \langle 1, 2 \rangle$. You get $\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||} = \frac{16}{\sqrt{52}\sqrt{5}} = \frac{16}{2\sqrt{65}}$. Therefore $\theta = \arccos\left(\frac{8}{\sqrt{65}}\right) \approx 0.1244$ radians or 7.125^o.

4 (c) $\begin{array}{l} x = t - 2,\ y = -t + 5 \\ x = -3s + 1,\ y = 3s - 4 \end{array}$

You need to determine if there are values for t and s where (x, y) are the same for both lines. So we need to solve these two equations for the two unknown values of s and t.

$$\begin{array}{rcl} t-2 &=& -3s+1 \\ -t+5 &=& 3s-4 \end{array} \sim \begin{array}{rcl} t+3s &=& 3 \\ -t-3s &=& -9 \end{array} \sim \begin{bmatrix} 1 & 3 \\ -1 & -3 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 3 \\ -9 \end{bmatrix}$$

The determinant is -3 + 3 = 0 and there is not a unique solution. You can perform Gaussian Elimination from here but you should notice that the two lines are parallel but do not go through the same point. Therefore there is no solution and no point of intersection.

5a Using equation (2.25) for a directed line segment in 3D,
$x(t) = 3 + t(-2 - 3) = 3 - 5t$
$y(t) = 2 + t(-1 - 2) = 2 - 3t$
$z(t) = 1 + t(1 - 1) = 1$
for $t \in [0, 1]$

5c Using equation (2.26) for a directed line segment in 2D,
$x(t) = 1 + t(-2 - 1) = 1 - 3t$
$y(t) = 2 + t(5 - 2) = 2 + 3t$
for $t \in [0, 1]$

6a You need to determine if there are values for t and s where (x, y) are the same for both lines. So we need to solve these two equations for the two unknown values of s and t.

$$\begin{array}{rrcll} & \overrightarrow{PQ} & = & \overrightarrow{RS} & \\ x: & 1+t(10-1) & = & 3+s(5-3) & \text{can be set up as a system of equations} \\ y: & 8+t(2-8) & = & 1+s(11-1) & \end{array}$$

$$\begin{array}{rcl} 1+9t & = & 3+2s \\ 8-6t & = & 1+10s \end{array} \sim \begin{array}{rcl} 9t-2s & = & 2 \\ -6t-10s & = & -7 \end{array} \sim \begin{bmatrix} 9 & -2 \\ -6 & -10 \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} 2 \\ -7 \end{bmatrix}$$

Solving this for t and s you get $t = 1/3$ and $s = 1/2$. Since both of these are between 0 and 1, there is a hit. The point of collision occurs at
$P + t(\overrightarrow{PQ}) = (1,8) + 1/3(9,-6) = (4,6)$ or $R + s(\overrightarrow{RS}) = (3,1) + 1/2(2,10) = (4,6)$.
So there is a hit at **(4,6)**.

Chapter 2.6

1a We need a point on the plane (we have three) and a normal vector. To get the normal vector $\mathbf{n}$ we set $\mathbf{n} = \overrightarrow{PQ} \times \overrightarrow{PR}$ which will be orthogonal to both vectors that form the plane. In our case, $\overrightarrow{PQ} = \langle 1,-1,1\rangle$, $\overrightarrow{PR} = \langle -2,2,2\rangle$, and

$$\mathbf{n} = \overrightarrow{PQ} \times \overrightarrow{PR} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & -1 & 1 \\ -2 & 2 & 2 \end{vmatrix} = -4\mathbf{i} - 4\mathbf{j} + 0\mathbf{k} = \langle -4,-4,0\rangle\,.$$

Using the point $P(0,1,2)$ as the point,

$$\begin{array}{rcll} -4(x-0)-4(y-1)+0(z-2) & = & 0 & \text{standard equation} \\ -4x-4y+4 & = & 0 & \text{intermediate step. now } \div \text{ -4} \\ x+y-1 & = & 0 & \text{general equation: simplest form} \end{array}$$

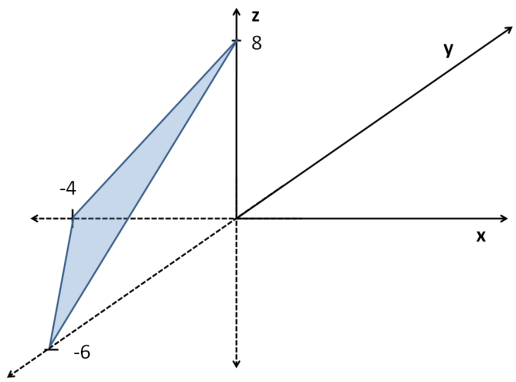
Note: Any multiple of this last equation is also an equation for the plane.

1c We have a point, now we need a normal vector. Since the z-axis is normal to the xy-plane, so it must be normal to our desired plane. The easiest vector parallel to the z-axis is $\mathbf{n} = \langle 0,0,1\rangle$.

$$\begin{array}{rcll} 0(x-1)+0(y-2)+1(z-3) & = & 0 & \text{standard equation} \\ z-3 & = & 0 & \text{intermediate step and general equation} \end{array}$$

Note: This is just the xy-plane shifted up (along the z-axis) 3 units. You might have been able to come up with the answer without using the standard procedure described above.

2a



$6x + 4y - 3z + 24 = 0$

x-intercept: $6x + 24 = 0 \to x = -4$

y-intercept: $4y + 24 = 0 \to y = -6$

z-intercept: $-3z + 24 = 0 \to z = 8$

3a The normal to the first plane is $\mathbf{n}_1 = \langle 1, 2, -2 \rangle$, and the normal to the second plane is $\mathbf{n}_2 = \langle 3, 0, -3 \rangle$, the angle of intersection is given by

$$\theta = \arccos\left(\frac{|\mathbf{n_1} \cdot \mathbf{n_2}|}{||\mathbf{n_1}||\ ||\mathbf{n_2}||}\right) = \arccos\left(\frac{9}{\sqrt{9}\,\sqrt{18}}\right) = \arccos\left(\frac{9}{3\ (3)\sqrt{2}}\right) =$$

$$\theta = \arccos\left(\frac{1}{\sqrt{2}}\right) = \arccos\left(\frac{\sqrt{2}}{2}\right) = \frac{\pi}{4} = 45^o$$

3b Method 1: We must find all points on the intersection of both planes. This means we must find values of x, y and z that satisfy both plane equations or

$$\begin{array}{rcl} x + 2y - 2z &=& 4 \\ 3x + 0y - 3z &=& 6 \end{array} \quad \sim \quad \begin{bmatrix} 1 & 2 & -2 \\ 3 & 0 & -3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Expressing these equations in augmented matrix form, you can use Gaussian Elimination to solve for the variables. You will get infinitely many solutions provided the planes actually intersect.

$$\begin{array}{rcl} x + 2y - 2z &=& 4 \\ 3x + 0y - 3z &=& 6 \end{array} \to \left[\begin{array}{ccc|c} 1 & 2 & -2 & 4 \\ 3 & 0 & -3 & 6 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & -2 & 4 \\ 0 & -6 & 3 & -6 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & -2 & 4 \\ 0 & 2 & -1 & 2 \end{array}\right]$$

You can see we will get infinitely many solutions. Looking at the last equation you see that $2y - z = 2$. If we let y be the free variable (to avoid fractions), then this equation becomes $z = -2 + 2y$. Looking at the first equation
$x + 2y - 2z = 4 \iff x + 2y - 2(-2 + 2y) = 4 \iff x + 2y + 4 - 4y = 4 \iff x = 2y$. Now let $y = t$ you get

$$x = 2t, \qquad y = t, \qquad z = -2 + 2t, \quad \text{for} \quad t \in (-\infty, \infty)$$

Notice, this is the line through $(0, 0, -2)$ with direction vector $\langle 2, 1, 2 \rangle$.

Method 2: (using MATLAB® calculations)

We know that the direction vector is given by $\mathbf{v} = \mathbf{n}_1 \times \mathbf{n}_2$;

$\mathbf{v} = \mathbf{n}_1 \times \mathbf{n}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 2 & -2 \\ 3 & 0 & -3 \end{vmatrix} = -6\mathbf{i} - 3\mathbf{j} - 6\mathbf{k} = \langle -6, -3, -6 \rangle$. Since any vector parallel to this will work,

you can divide this by -3 to get
$\mathbf{v} = \langle 2, 1, 2\rangle$ (same direction vector from Method 1).

Now we must find a point on the intersection of both planes. This means we must find values of x, y and z that satisfy both plane equations or

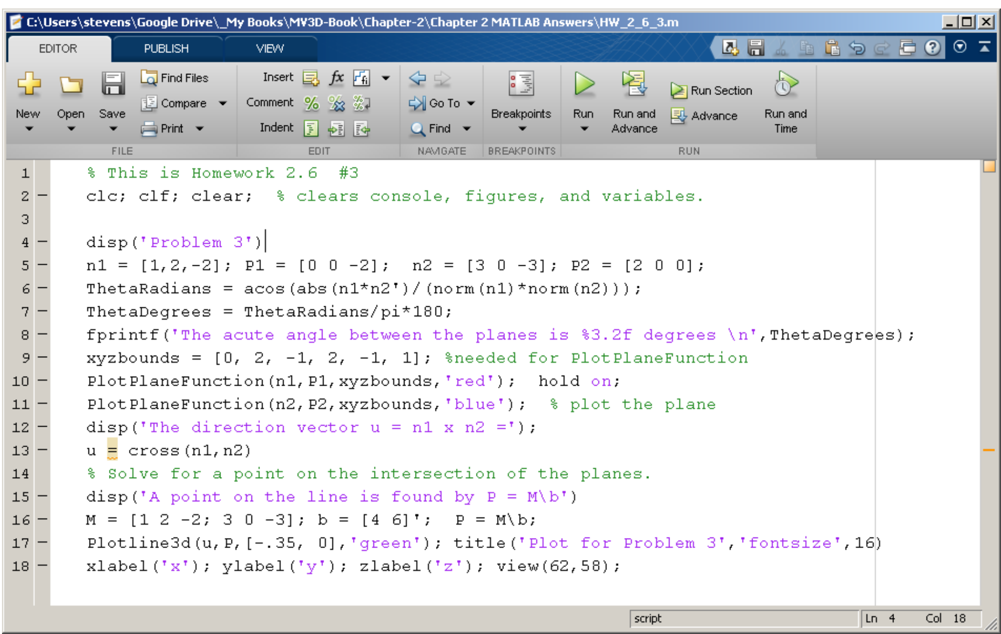
$$\begin{aligned} x + 2y - 2z &= 4 \\ 3x + 0y - 3z &= 6 \end{aligned} \rightarrow \begin{bmatrix} 1 & 2 & -2 \\ 3 & 0 & -3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}.$$

Now let $M = \begin{bmatrix} 1 & 2 & -2 \\ 3 & 0 & -3 \end{bmatrix}$ and $b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$ and have MATLAB® solve this system by $P = M\backslash b$. There are infinitely many but MATLAB® returns one. It returns $P(0, 0, -2)$ which is coincidentally the same point from Method 1. Now we have a direction vector $\mathbf{v} = \langle 2, 1, 2\rangle$ and a point (0, 0, -2). So the parametric equations for the line of intersection are

$$x = 2s, \qquad y = s, \qquad z = -2 + 2s, \quad \text{for} \quad s \in (-\infty, \infty)$$

Notice, this is the line through $(0, 0, -2)$ with direction vector $\langle 2, 1, 2\rangle$. There are infinitely many ways to express the same line, but they all have a direction vector parallel to $\langle 2, 1, 2\rangle$.

3c Here is the code used to find the angle, the direction vector, a point, and create the graph:




```
% This is Homework 2.6  #3
clc; clf; clear;  % clears console, figures, and variables.

disp('Problem 3')
n1 = [1,2,-2]; P1 = [0 0 -2];  n2 = [3 0 -3]; P2 = [2 0 0];
ThetaRadians = acos(abs(n1*n2')/(norm(n1)*norm(n2)));
ThetaDegrees = ThetaRadians/pi*180;
fprintf('The acute angle between the planes is %3.2f degrees \n',ThetaDegrees);
xyzbounds = [0, 2, -1, 2, -1, 1]; %needed for PlotPlaneFunction
PlotPlaneFunction(n1,P1,xyzbounds,'red');  hold on;
PlotPlaneFunction(n2,P2,xyzbounds,'blue');  % plot the plane
disp('The direction vector u = n1 x n2 =');
u = cross(n1,n2)
% Solve for a point on the intersection of the planes.
disp('A point on the line is found by P = M\b')
M = [1 2 -2; 3 0 -3]; b = [4 6]';  P = M\b;
Plotline3d(u,P,[-.35, 0],'green'); title('Plot for Problem 3','fontsize',16)
xlabel('x'); ylabel('y'); zlabel('z'); view(62,58);
```

5a The distance is given by $D = ||\text{proj}_{\mathbf{n}}\overrightarrow{PQ}|| = \dfrac{|\overrightarrow{PQ} \cdot \mathbf{n}|}{||\mathbf{n}||}$
where $\mathbf{n}$ is the normal to the plane, Q is the point of interest, and P is any point on the plane.

Here, $\mathbf{n} = \langle 6, -3, 2 \rangle$, $Q = (1, -1, 1)$, and we'll let $P = (0, 0, 4)$. Therefore $\overrightarrow{PQ} = \langle 1, -1, -3 \rangle$.
So, $D = \dfrac{|\,\langle 1, -1, -3 \rangle \cdot \langle 6, -3, 2 \rangle\,|}{||\,\langle 6, -3, 2 \rangle\,||} = \dfrac{|3|}{\sqrt{49}} = \dfrac{3}{7} \approx 0.43$

6a The sphere has center at $Q(1, -2, 0)$ and radius $= 2$. So we check the distance from the center point to the plane. This distance is

$$D = ||\text{proj}_{\mathbf{n}} \overrightarrow{PQ}|| = \frac{|\overrightarrow{PQ} \cdot \mathbf{n}|}{||\mathbf{n}||}$$

where $\mathbf{n}$ is the normal to the plane, and P is any point on the plane. Here, $\mathbf{n} = \langle 2, -3, 1 \rangle$ and we'll let $P = (0, 0, 2)$. Therefore $\overrightarrow{PQ} = \langle 1, -2, -2 \rangle$.

So,

$$D = \frac{|\overrightarrow{PQ} \cdot \mathbf{n}|}{||\mathbf{n}||} = \frac{|\,\langle 1, -2, -2 \rangle \cdot \langle 2, -3, 1 \rangle\,|}{||\,\langle 2, -3, 1 \rangle\,||} = \frac{6}{\sqrt{14}} \approx 1.60$$

Since the radius of the sphere is 2 and the distance from the point to the plane ≈ 1.60 we conclude that the sphere and the plane do intersect.

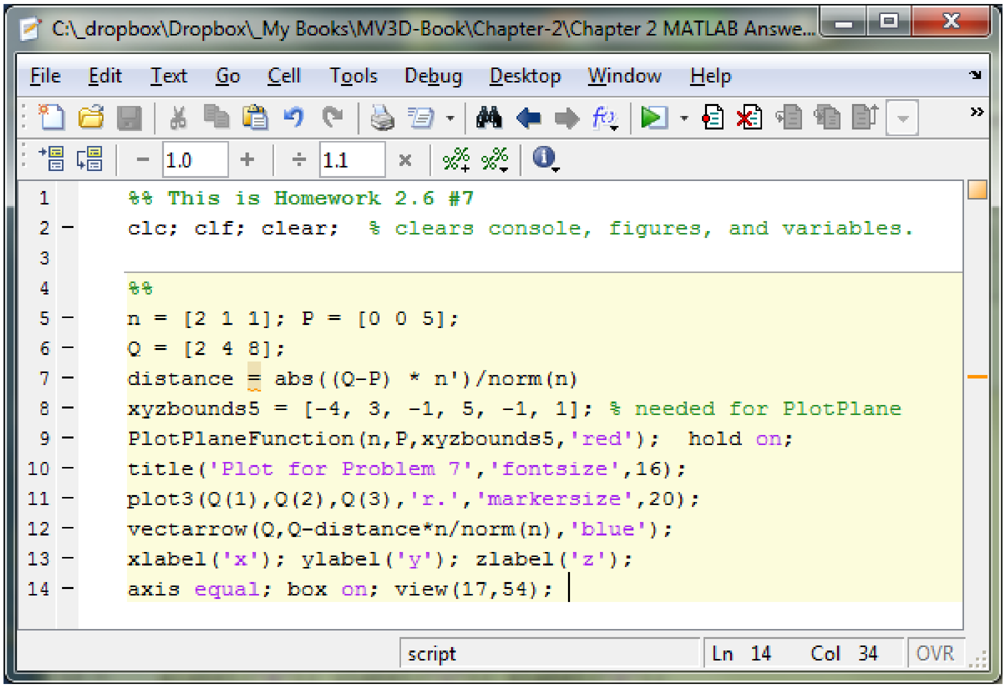
7 We will use the formula for the distance between a point Q and the plane with normal vector $\mathbf{n}$ by

$$D = ||\text{proj}_{\mathbf{n}} \overrightarrow{PQ}|| = \frac{|\overrightarrow{PQ} \cdot \mathbf{n}|}{||\mathbf{n}||}$$

where P is a point in the plane. In our case, $\mathbf{n} = \langle 2, 1, 1 \rangle$, $Q = (2, 4, 8)$, and we'll let $P = (0, 0, 5)$. Therefore, $\overrightarrow{PQ} = \langle -2, -4, -3 \rangle$. And

$$D = \frac{|\overrightarrow{PQ} \cdot \mathbf{n}|}{||\mathbf{n}||} = \frac{|-11|}{\sqrt{6}} = \frac{11}{\sqrt{16}} \approx 4.49$$

Here is the code I used to get the distance, draw the point, the plane, and the directed line segment:




```
%% This is Homework 2.6 #7
clc; clf; clear;  % clears console, figures, and variables.

%%
n = [2 1 1]; P = [0 0 5];
Q = [2 4 8];
distance = abs((Q-P) * n')/norm(n)
xyzbounds5 = [-4, 3, -1, 5, -1, 1]; % needed for PlotPlane
PlotPlaneFunction(n,P,xyzbounds5,'red');  hold on;
title('Plot for Problem 7','fontsize',16);
plot3(Q(1),Q(2),Q(3),'r.','markersize',20);
vectarrow(Q,Q-distance*n/norm(n),'blue');
xlabel('x'); ylabel('y'); zlabel('z');
axis equal; box on; view(17,54);
```

8 We'll use the formula for the distance between a point and a plane:

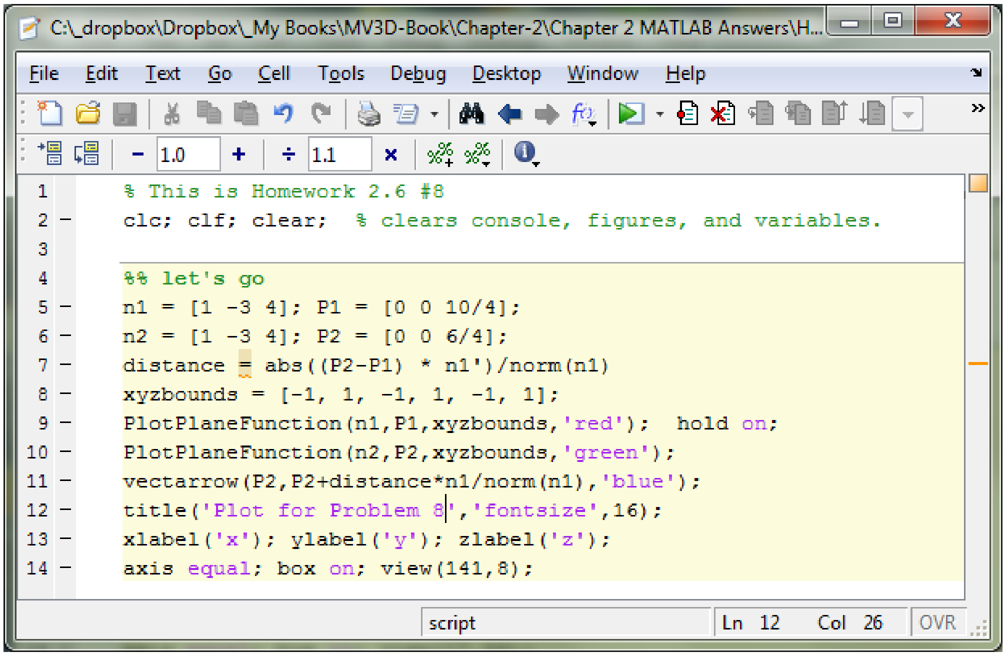
$$D = ||\text{proj}_{\mathbf{n}}\overrightarrow{PQ}|| = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||}.$$

There are many ways to set this up. I'll choose some point (Q) from the second plane and find the distance from that point to the first plane. The easiest selection for Q is $Q = (6,0,0)$. The normal to the first plane is $\mathbf{n} = \langle 1,-3,4\rangle$ and a nice point P from the first plane is $P = (10,0,0)$. Therefore, $\overrightarrow{PQ} = \langle 4,0,0\rangle$.

So,

$$D = \frac{|\overrightarrow{PQ}\cdot\mathbf{n}|}{||\mathbf{n}||} = \frac{|4|}{\sqrt{26}} \approx 0.78.$$

Here is the code I used to find the distance, sketch the planes and the directed line segment between them:



Chapter 2.7

1 Doing this by hand, you first get the parametric equations for the line.
The direction vector $\mathbf{v} = \overrightarrow{MQ} = \langle 20, 20, 20\rangle$ and we'll use $M(-10, -10, -10)$ as the point.
Line: $x = -10 + 20t, \quad y = -10 + 20t, \quad z = -10 + 20t$
Plugging these expressions for x, y, and z into the equation for the plane you get
$2(-10 + 20t) + (-10 + 20t) - (-10 + 20t) + 1 = 0 \quad \rightarrow \quad -19 + 40t = 0 \quad \rightarrow \quad t = \frac{19}{40} = 0.475.$
So the line hits the plane when $t = 0.475$. The point of intersection is found but subbing this value of t into the parametric equations for the line.
$x = -10 + 20(19/40), \quad y = -10 + 20(19/40), \quad z = -10 + 20(19/40)$
So the point $P = (-0.5,\ -0.5,\ -0.5)$.

Using the shortcut, the point of intersection is given by equation (2.32) $P = P_L + t\,\mathbf{v}$
where t is given by equation (2.31) $t = \frac{\mathbf{n}\cdot\langle P_P - P_L\rangle}{\mathbf{n}\cdot\mathbf{v}}$,
$\mathbf{n}$ is a normal to the plane, P_p is a point on the plane, $\mathbf{v}$ is a direction vector for the line, and P_L is a point on the line. So,

$$\begin{aligned} \mathbf{n} &= \langle 2, 1, -1\rangle \\ P_p &= (0, 0, 1) \\ \mathbf{v} &= \overrightarrow{MQ} = \langle 20, 20, 20\rangle \\ P_L &= M = (-10, -10, -10) \end{aligned}$$

$$t = \frac{\mathbf{n} \cdot \langle P_P - P_L \rangle}{\mathbf{n} \cdot \mathbf{v}} = \frac{\langle 2, 1, -1 \rangle \cdot \langle 10, 10, 11 \rangle}{\langle 2, 1, -1 \rangle \cdot \langle 20, 20, 20 \rangle} = \frac{19}{40} = 0.475$$

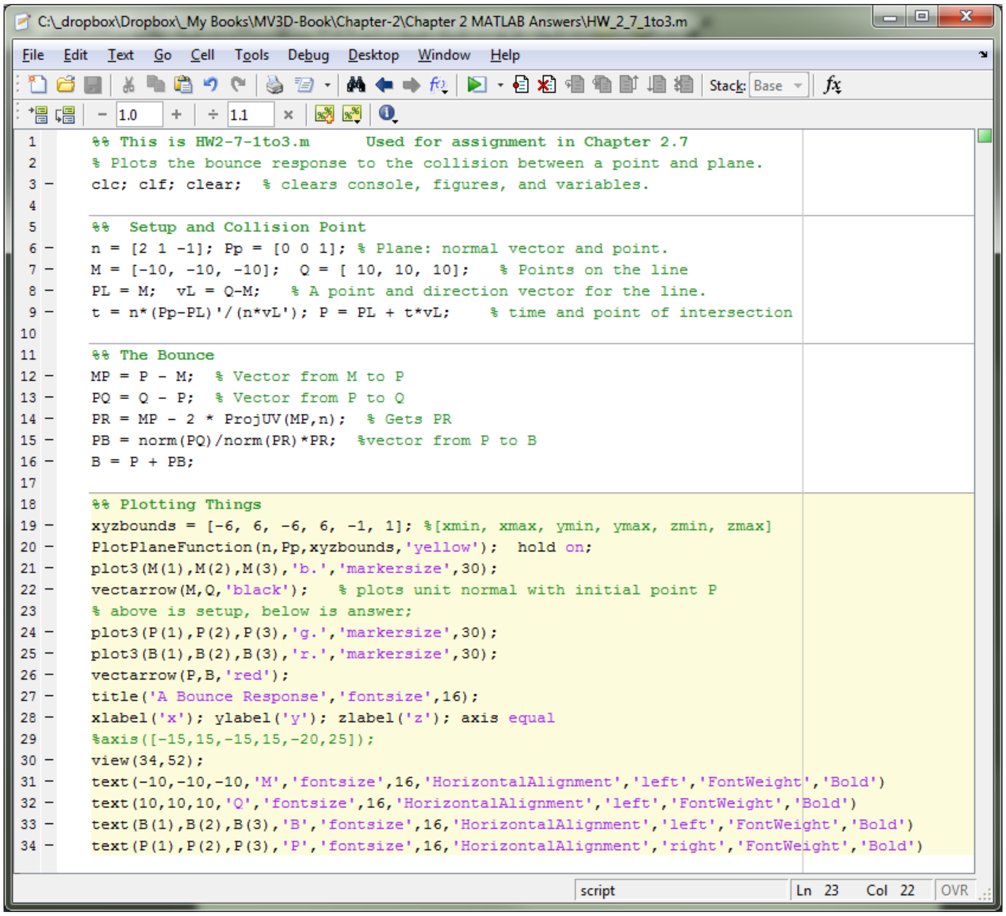
and

$$P \;=\; P_L + t\,\mathbf{v} = (-10, -10, -10) + \frac{19}{40}\,\langle 20, 20, 20 \rangle = (-0.5, -0.5, -0.5)$$

2 To get things started, we have $P = (-0.5,\ -0.5,\ -0.5)$ from the previous problem. Therefore, you get $\overrightarrow{MP} = \langle 9.5,\ 9.5,\ 9.5 \rangle$ and $\overrightarrow{PQ} = \langle 10.5,\ 10.5,\ 10.5, \rangle$. Additionally, $n = \langle 2,\ 1,\ -1 \rangle$ from the previous problem as well. Now, use equations (2.33) - (2.35) to get the final position of the ball (B).

$$\begin{aligned}
\overrightarrow{PR} &= \overrightarrow{MP} - 2\ \mathrm{Proj}_{\mathbf{n}} \overrightarrow{MP} \\
&= \langle 9.5,\ 9.5,\ 9.5 \rangle - 2\ \frac{\langle 9.5,\ 9.5,\ 9.5 \rangle \cdot \langle 2,\ 1,\ -1 \rangle}{||\,\langle 2,\ 1,\ -1 \rangle\,||^2}\ \langle 2,\ 1,\ -1 \rangle \\
&= \langle 9.5,\ 9.5,\ 9.5 \rangle - 2\ \frac{19}{6}\ \langle 2,\ 1,\ -1 \rangle = \left\langle \frac{-19}{6},\ \frac{19}{6},\ \frac{95}{6} \right\rangle \approx \langle -3.17,\ 3.17,\ 15.83 \rangle \\
\overrightarrow{PB} &= \frac{||\overrightarrow{PQ}||}{||\overrightarrow{PR}||}\ \overrightarrow{PR} = \langle -3.5,\ 3.5,\ 17.5 \rangle \\
B &= P + \overrightarrow{PB} = (-0.5,\ -0.5,\ -0.5) + \langle -3.5,\ 3.5,\ 17.5 \rangle = (-4,\ 3,\ 17)
\end{aligned}$$

3 Below is the code for the calculations for the first two problems as well as the resulting graph.



```
%% This is HW2-7-1to3.m      Used for assignment in Chapter 2.7
% Plots the bounce response to the collision between a point and plane.
clc; clf; clear;  % clears console, figures, and variables.

%%  Setup and Collision Point
n = [2 1 -1]; Pp = [0 0 1]; % Plane: normal vector and point.
M = [-10, -10, -10];  Q = [ 10, 10, 10];   % Points on the line
PL = M;  vL = Q-M;   % A point and direction vector for the line.
t = n*(Pp-PL)'/(n*vL'); P = PL + t*vL;    % time and point of intersection

%% The Bounce
MP = P - M;  % Vector from M to P
PQ = Q - P;  % Vector from P to Q
PR = MP - 2 * ProjUV(MP,n);  % Gets PR
PB = norm(PQ)/norm(PR)*PR;  %vector from P to B
B = P + PB;

%% Plotting Things
xyzbounds = [-6, 6, -6, 6, -1, 1]; %[xmin, xmax, ymin, ymax, zmin, zmax]
PlotPlaneFunction(n,Pp,xyzbounds,'yellow');  hold on;
plot3(M(1),M(2),M(3),'b.','markersize',30);
vectarrow(M,Q,'black');   % plots unit normal with initial point P
% above is setup, below is answer;
plot3(P(1),P(2),P(3),'g.','markersize',30);
plot3(B(1),B(2),B(3),'r.','markersize',30);
vectarrow(P,B,'red');
title('A Bounce Response','fontsize',16);
xlabel('x'); ylabel('y'); zlabel('z'); axis equal
%axis([-15,15,-15,15,-20,25]);
view(34,52);
text(-10,-10,-10,'M','fontsize',16,'HorizontalAlignment','left','FontWeight','Bold')
text(10,10,10,'Q','fontsize',16,'HorizontalAlignment','left','FontWeight','Bold')
text(B(1),B(2),B(3),'B','fontsize',16,'HorizontalAlignment','left','FontWeight','Bold')
text(P(1),P(2),P(3),'P','fontsize',16,'HorizontalAlignment','right','FontWeight','Bold')
```

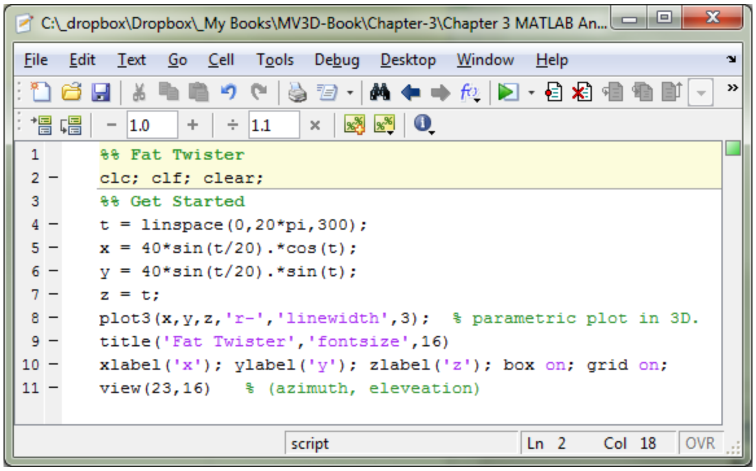
Chapter 3.1

1a The x-radius is 2, and the y-radius is 1
$x(t) = 2 + 2\cos(t), \quad y(t) = 3 + \sin(t), \qquad t \in [-\pi, 0]$ or $t \in [\pi, 2\pi]$
There are other answers that trace out the exact same curve.

2 (a) You want a circle in the xy-plane that increases then decreases in radius as t increases and z increases as t increases. There are about 10 rotations so $t \in [0, 20\pi]$.

$$x(t) = 40\sin(t/20)\cos(t), \quad y(t) = 40\sin(t/20)\sin(t), \quad z = t, \quad t \in [0, 20\pi]$$

Here, the radius of the circle goes like $40\sin(t/20)$ which is the first half of the sine wave.

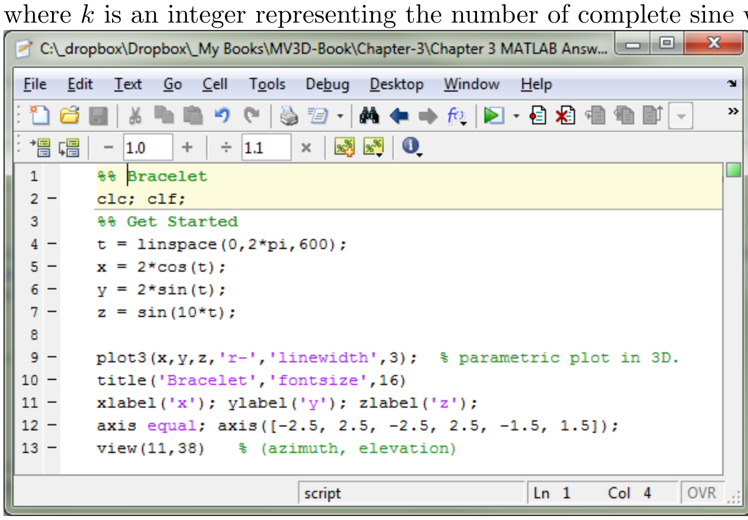


```
%% Fat Twister
clc; clf; clear;
%% Get Started
t = linspace(0,20*pi,300);
x = 40*sin(t/20).*cos(t);
y = 40*sin(t/20).*sin(t);
z = t;
plot3(x,y,z,'r-','linewidth',3);  % parametric plot in 3D.
title('Fat Twister','fontsize',16)
xlabel('x'); ylabel('y'); zlabel('z'); box on; grid on;
view(23,16)    % (azimuth, eleveation)
```

(b) You want x and y to create a circle in the xy-plane and let z oscillate more quickly.

$$x(t) = 2\cos(t), \quad y(t) = 2\sin(t) \quad z = \sin(kt), \quad t \in [0, 2\pi]$$

where k is an integer representing the number of complete sine waves.

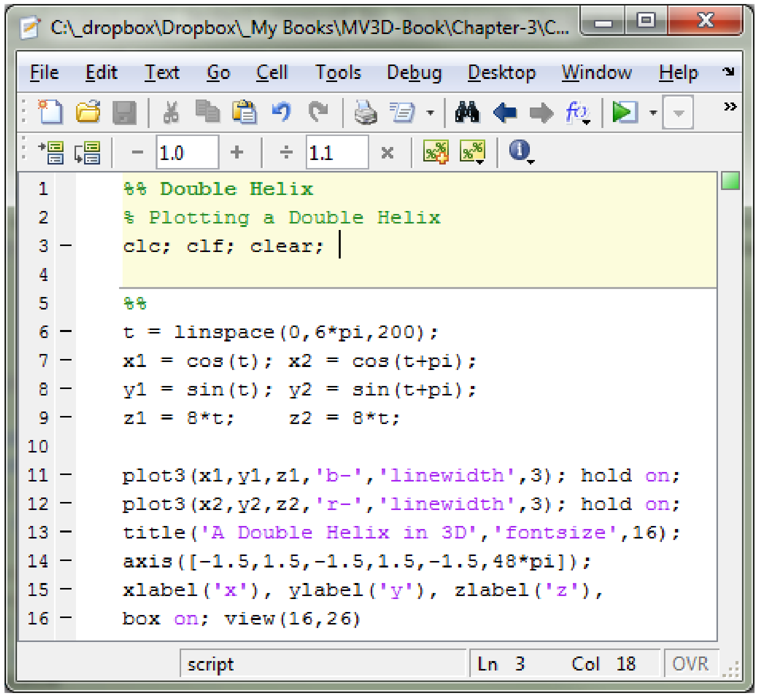


```
%% Bracelet
clc; clf;
%% Get Started
t = linspace(0,2*pi,600);
x = 2*cos(t);
y = 2*sin(t);
z = sin(10*t);

plot3(x,y,z,'r-','linewidth',3);  % parametric plot in 3D.
title('Bracelet','fontsize',16)
xlabel('x'); ylabel('y'); zlabel('z');
axis equal; axis([-2.5, 2.5, -2.5, 2.5, -1.5, 1.5]);
view(11,38)    % (azimuth, elevation)
```

4 Blue Helix:
$x(t) = \cos(t)$
$y(t) = \sin(t)$
$z(t) = 8t,$
for $t \in [0, 6\pi]$

Red Helix:
$x(t) = \cos(t + \pi)$
$y(t) = \sin(t + \pi)$
$z(t) = 8t,$
for $t \in [0, 6\pi]$

Others are possible


C:_dropbox\Dropbox_My Books\MV3D-Book\Chapter-3\C...

File Edit Text Go Cell Tools Debug Desktop Window Help

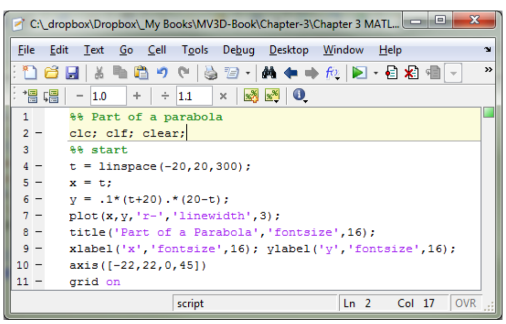
```
%% Double Helix
% Plotting a Double Helix
clc; clf; clear;

%%
t = linspace(0,6*pi,200);
x1 = cos(t); x2 = cos(t+pi);
y1 = sin(t); y2 = sin(t+pi);
z1 = 8*t;    z2 = 8*t;

plot3(x1,y1,z1,'b-','linewidth',3); hold on;
plot3(x2,y2,z2,'r-','linewidth',3); hold on;
title('A Double Helix in 3D','fontsize',16);
axis([-1.5,1.5,-1.5,1.5,-1.5,48*pi]);
xlabel('x'), ylabel('y'), zlabel('z'),
box on; view(16,26)
```

script Ln 3 Col 18 OVR

5a

$$\mathbf{v}(t) = \left\langle t,\ \frac{1}{10}(t+20)(20-t) \right\rangle \quad \text{for} \quad t \in [-20, 20]$$

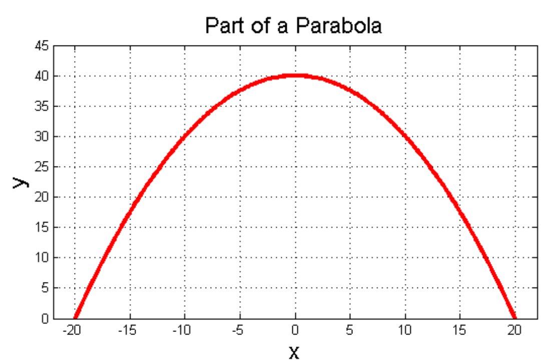
Below is the code


C:_dropbox\Dropbox_My Books\MV3D-Book\Chapter-3\Chapter 3 MATL...

File Edit Text Go Cell Tools Debug Desktop Window Help

```
%% Part of a parabola
clc; clf; clear;
%% start
t = linspace(-20,20,300);
x = t;
y = .1*(t+20).*(20-t);
plot(x,y,'r-','linewidth',3);
title('Part of a Parabola','fontsize',16);
xlabel('x','fontsize',16); ylabel('y','fontsize',16);
axis([-22,22,0,45])
grid on
```

script Ln 2 Col 17 OVR



Chapter 3.2

1a In order to find the velocity function we differentiate the position vector with respect to t.

$$\mathbf{r}(t) = \langle\, 2\cos(t), \quad 8\sin(t)\,\rangle \quad \rightarrow \quad \mathbf{v}(t) = \mathbf{r}'(t) = \langle\, -2\sin(t), \quad 8\cos(t)\,\rangle\,.$$

1b We really need t, not x and y. If you sketch the ellipse and this point, it is pretty obvious that $t = 0$. However, you can use the arctan function to confirm this. We know that $x = 2\cos(t) = 2$ and $y = 8\sin(t) = 0$.

Therefore

$$\begin{aligned}2\cos(t) &= 2 \quad \rightarrow \quad \cos(t) = 1\\ 8\sin(t) &= 0 \quad \rightarrow \quad \sin(t) = 0\\ \tan(t) &= \frac{\sin(t)}{\cos(t)} = \frac{0}{1}\\ t &= \texttt{atan2}(0,1) = 0\end{aligned}$$

So, $t = 0$ and the velocity vector is

$$v(0) = \langle\ -2\sin(0), \quad 8\cos(0)\ \rangle = \langle\ \mathbf{0}, \mathbf{8}\ \rangle$$

.

1c We really need t, not x and y. If you sketch the ellipse and this point, it is pretty obvious that t is 270° or -90° which is $3\pi/2$ or $-\pi/2$ respectively. However, you can use the arctan function to confirm this. We know that $x = 2\cos(t) = 0$ and $y = 8\sin(t) = -8$. Therefore

$$\begin{aligned}2\cos(t) &= 0 \quad \rightarrow \quad \cos(t) = 0\\ 8\sin(t) &= -8 \quad \rightarrow \quad \sin(t) = -1\\ t &= \texttt{atan2}(-1,0) = -1.570796 = -\pi/2\end{aligned}$$

So, $t = -\pi/2$ and the velocity vector is

$$v(-\pi/2) = \langle\ -2\sin(-\pi/2), \quad 8\cos(-\pi/2)\ \rangle = \langle\ \mathbf{2}, \mathbf{0}\ \rangle$$

Technical Note: The arctan function returned $t = -\pi/2$ but that is not in the specified domain of our function. The value of t within the specified domain would be $t = -\pi/2 + 2\pi = 3\pi/2$. Since our parametric functions are periodic with a period of 2π, the value of t returned by the arctan function gets us the same position and velocity vector. I.e., it still works.

1d We really need t, not x and y. If you sketch the ellipse and the point, it is not obvious what t is. However, you can use the arctan function to find t. We know that $x = 2\cos(t) = -\sqrt{2}$ and $y = 8\sin(t) = 4\sqrt{2}$. Therefore

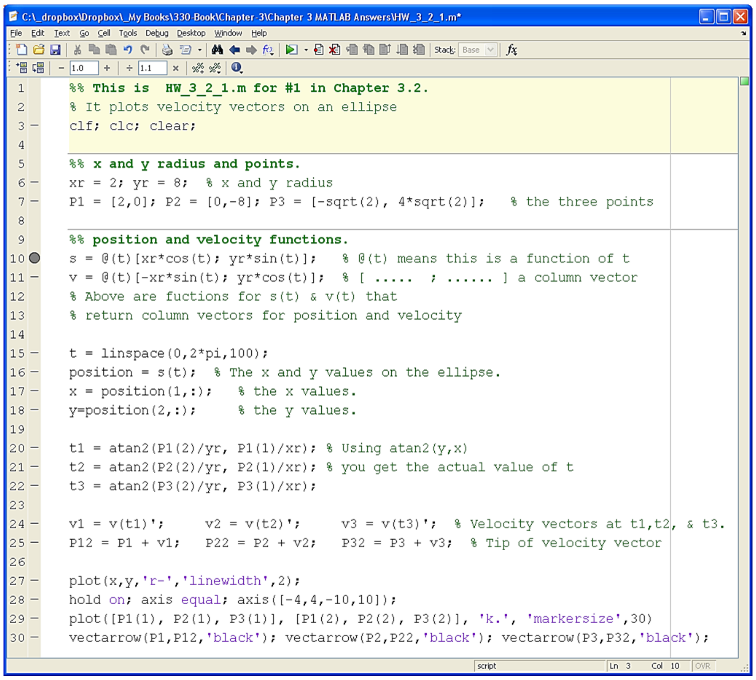
$$\begin{aligned}2\cos(t) &= -\sqrt{2} \quad \rightarrow \quad \cos(t) = -\frac{\sqrt{2}}{2}\\ 8\sin(t) &= 4\sqrt{2} \quad \rightarrow \quad \sin(t) = \frac{\sqrt{2}}{2}\\ t &= \texttt{atan2}(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}) = 2.3562 = \frac{3\pi}{4}\end{aligned}$$

So, $t = 3\pi/4$ and the velocity vector is

$$v(3\pi/4) = \langle\ -2\sin(3\pi/4), \quad 8\cos(3\pi/4)\ \rangle$$

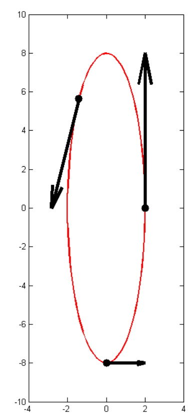
$$= \left\langle -2\,\frac{\sqrt{2}}{2},\, 8\,\frac{-\sqrt{2}}{2} \right\rangle$$
$$= \left\langle -\sqrt{2}, -4\sqrt{2} \right\rangle \approx \langle\, \mathbf{-1.41, -5.66}\, \rangle.$$

1e Here is the code and the graph.



C:_dropbox\Dropbox_My Books\330-Book\Chapter-3\Chapter 3 MATLAB Answers\HW_3_2_1.m*

```
%% This is  HW_3_2_1.m for #1 in Chapter 3.2.
% It plots velocity vectors on an ellipse
clf; clc; clear;

%% x and y radius and points.
xr = 2; yr = 8;  % x and y radius
P1 = [2,0]; P2 = [0,-8]; P3 = [-sqrt(2), 4*sqrt(2)];   % the three points

%% position and velocity functions.
s = @(t)[xr*cos(t); yr*sin(t)];   % @(t) means this is a function of t
v = @(t)[-xr*sin(t); yr*cos(t)];  % [ .....  ; ...... ] a column vector
% Above are fuctions for s(t) & v(t) that
% return column vectors for position and velocity

t = linspace(0,2*pi,100);
position = s(t);  % The x and y values on the ellipse.
x = position(1,:);   % the x values.
y=position(2,:);     % the y values.

t1 = atan2(P1(2)/yr, P1(1)/xr); % Using atan2(y,x)
t2 = atan2(P2(2)/yr, P2(1)/xr); % you get the actual value of t
t3 = atan2(P3(2)/yr, P3(1)/xr);

v1 = v(t1)';     v2 = v(t2)';     v3 = v(t3)';  % Velocity vectors at t1,t2, & t3.
P12 = P1 + v1;   P22 = P2 + v2;   P32 = P3 + v3;  % Tip of velocity vector

plot(x,y,'r-','linewidth',2);
hold on; axis equal; axis([-4,4,-10,10]);
plot([P1(1), P2(1), P3(1)], [P1(2), P2(2), P3(2)], 'k.', 'markersize',30)
vectarrow(P1,P12,'black'); vectarrow(P2,P22,'black'); vectarrow(P3,P32,'black');
```



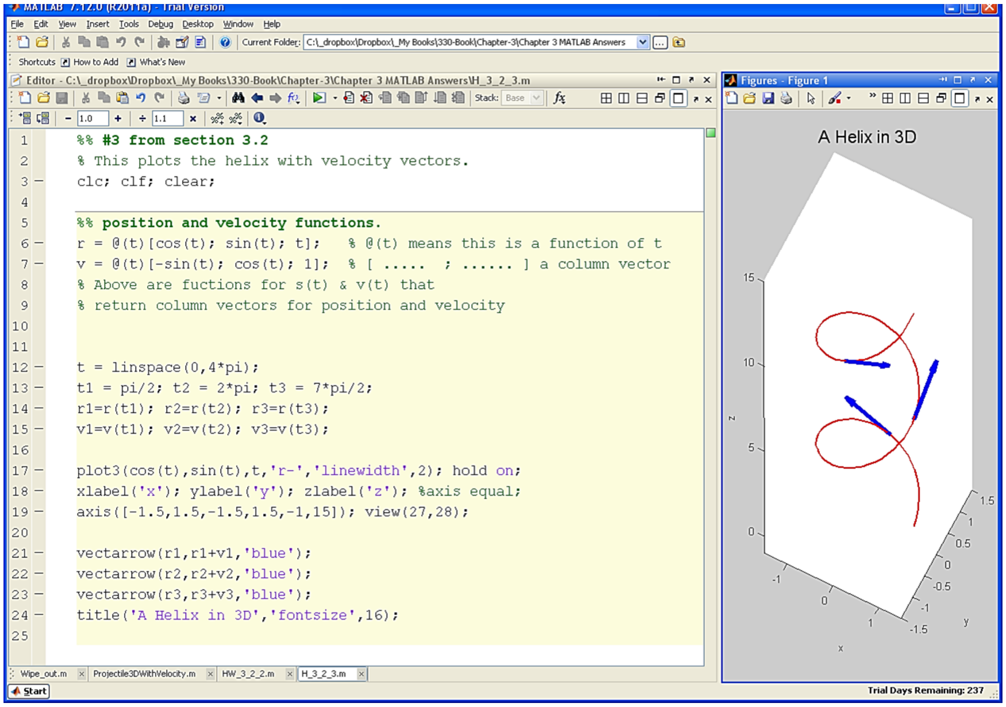
3a

$$\mathbf{v}(t) = \langle -\sin(t), \cos(t), 1 \rangle$$

$$\mathbf{a}(t) = \langle -\cos(t), -\sin(t), 0 \rangle$$

3b There is no acceleration in the z-direction, the velocity is constant in that direction.

3c Here is the code and graph



5a Differentiating term-by-term, (using the product rule on $x(t)$),

$$\mathbf{v}(t) = \langle -t\sin(t) + \cos(t),\ 4\cos(4t),\ 6t \rangle$$

5b

$$\mathbf{v}(0) = \langle\ 1,\ 4, 0\ \rangle\,.$$

$$\text{speed} = \sqrt{1+16+0} = \sqrt{17}.$$

5c This is a little refresher in differentiation with respect to $x'(t)$ so let's start with that one

$$\begin{aligned} x'(t) &= -t\sin(t) + \cos(t) \\ x''(t) &= -t(\cos(t)) + \sin(t)(-1) - \sin(t) \\ &= -t\cos(t) - 2\sin(t), \end{aligned}$$

$$y'(t) = 4\cos(4t) \quad \rightarrow \quad y''(t) = -16\sin(4t),$$

and

$$z'(t) = 6t \quad \rightarrow \quad z''(t) = 6$$

Finally,

$$\mathbf{a}(t) = \mathbf{v}'(t) = \langle\ -t\cos(t) - 2\sin(t),\ -16\sin(4t),\ 6\ \rangle$$

6 $\mathbf{v}(t) = \left\langle 2,\ 2t,\ \frac{3}{2}t^2 \right\rangle$ but you need t. If the point is (4,5,4) this means that $2t = 4$ or $t = 2$. So, $\mathbf{v}(2) = \langle \mathbf{2},\ \mathbf{4},\ \mathbf{6} \rangle$

Chapter 3.3

1a Using equation (3.11): $\mathbf{r}(t) = \langle x(t),\ y(t) \rangle = \left\langle [v_o \cos(\theta)]t, \quad h + [v_o \sin(\theta)]t - \frac{1}{2}gt^2 \right\rangle$.
with $\theta = 0$, $h = 400$, $g = 32$, and $v_o = 60$ we get $\mathbf{r}(t) = \left\langle 60t, \quad 400 - 16t^2 \right\rangle = \langle x(t),\ y(t) \rangle$.

1b You get the height at $t = 2$ from $y(2) = 400 - 16 \cdot 2^2 =$**336 feet.**

1c Solving $y(t) = 0$ you get the ball hits the ground at $t^* = 5$, and plugging $t = 5$ into $x(t)$ gives the total distance traveled as $x(5) = \mathbf{300}$.

You can also do this use equations (3.12) (for t^*) and (3.13) for distance.

$$t^* = \frac{v_o \sin(\theta) + \sqrt{v_o^2 \sin^2(\theta) + 2gh}}{g} = \frac{0 + \sqrt{0 + 2(32)(400)}}{32} = \frac{\sqrt{25600}}{32} = 5 \text{ sec}$$

$$\text{total distance} = \frac{v_o^2 \cos(\theta)}{g} \left(\sin(\theta) + \sqrt{\sin^2(\theta) + \frac{2gh}{v_o^2}} \right) = 112.5\sqrt{\frac{25600}{3600}} = 300 \text{ ft.}$$

1d The speed on impact is the magnitude of the velocity vector at $t = 5$.

$$\begin{aligned} \mathbf{v}(t) = \mathbf{r}'(t) &= \langle 60, \quad -32t \rangle \\ \mathbf{v}(5) &= \langle 60, \quad -32(5) \rangle = \langle 60, \quad -160 \rangle \\ ||\mathbf{v}(5)|| &= \sqrt{60^2 + (-160)^2} = \sqrt{29200} \approx \mathbf{170.9} \text{ ft/sec.} \end{aligned}$$

3a Here, $h = 28$, $\theta = 30^o = \pi/6$, $v_o = 96$ ft/sec, and $g = 32$ feet/sec^2.
Note: $\cos(\pi/6) = \frac{\sqrt{3}}{2}$, $\sin(\pi/3) = \frac{1}{2}$
Plugging these values into the equation for when the ball hits the ground.

$$t^* = \frac{v_o \sin(\theta) + \sqrt{v_o^2 \sin^2(\theta) + 2gh}}{g}$$

$$t^* = \frac{96(1/2) + \sqrt{(96 * (1/2))^2 + 2 \cdot 32 \cdot 28}}{32} = \frac{48 + \sqrt{4096}}{32} = \frac{48 + 64}{32} = 3.5 \text{ seconds}$$

3b $\mathbf{r}(t) = \langle\ x(t),\ y(t)\ \rangle = \left\langle v_o \cos(\theta)\ t, \quad h + v_o \sin(\theta)\ t - 16t^2\ \right\rangle$

$\mathbf{r}(t) = \left\langle 96\cos(\pi/6)\ t, \quad 28 + 96\sin(\pi/6)\ t - 16t^2\ \right\rangle = \left\langle 96\frac{\sqrt{3}}{2}\ t, \quad 28 + 96\frac{1}{2}\ t - 16t^2 \right\rangle$

$\mathbf{r}(t) = \left\langle 48\sqrt{3}\ t, \quad 28 + 48\ t - 16t^2\ \right\rangle \approx \left\langle 83.1\ t, \quad 28 + 48\ t - 16t^2\ \right\rangle$ for $\quad t \in [0, t^*]$

3c $\mathbf{v}(t) = \mathbf{r}'(t) = \left\langle 48\sqrt{3}, \quad 48 - 32t\ \right\rangle \approx \langle 83.1, \quad 48 - 32t\ \rangle$ for $\quad t \in [0, t^*]$

3d The velocity vector when it hits the ground is $\mathbf{v}(t^*)$
$\mathbf{v}(3.5) = \left\langle 48\sqrt{3}, \quad 48 - 32(3.5) \right\rangle = \left\langle 48\sqrt{3}, \quad -64 \right\rangle \approx \langle 83.1, \quad -64 \rangle$
The speed is $||\mathbf{v}(3.5)|| = \sqrt{(48\sqrt{3})^2 + (-64)^2} = \sqrt{6912 + 4096} = \sqrt{11008} \approx 104.9$ ft/sec.

3e You get the time of maximum height by differentiating $y(t)$ and setting this equal to zero.
$y'(t) = 48 - 32t = 0 \rightarrow \tilde{t} = 48/32 = 1.5$ sec.
The actual height is $y(1.5) = 28 + 48(1.5) - 16(1.5)^2 = 64$ ft.

3f Set $y(t) = 28$.
$28 + 48\,t - 16t^2 = 28 \rightarrow 48t - 16t^2 = 0 \rightarrow 48t(t-3) = 0 \rightarrow t = 0 \quad \text{or} \quad t = 3.$
We disregard $t = 0$ and the projectile is again 28 feet high at $t = 3$ seconds.

5a Here, $h = 60$, $\phi = 30^o = \pi/6$, $\theta = 45^o = \pi/4$, $v_o = 100$ ft/sec, and $g = 32$ feet/sec^2.
Note: $\sin(\pi/6) = \frac{1}{2}$, $\cos(\pi/6) = \frac{\sqrt{3}}{2}$, $\cos(\pi/4) = \sin(\pi/4) = \frac{\sqrt{2}}{2}$.
Plugging these values into equation (3.24) for when the ball hits the ground we get

$$t^* = \frac{v_o \sin(\phi) + \sqrt{v_o^2 \sin^2(\phi) + 2gh}}{g} \approx 4.05 \text{ seconds}$$

5b Now that we have the total time the ball is in the air, we use equation (3.23) for the actual trajectory:

$$\begin{aligned}
\mathbf{r}(t) &= \langle\, x(t),\ y(t),\ z(t)\,\rangle \\
&= \left\langle v_o \cos(\theta)\cos(\phi)\,t, \quad v_o \sin(\theta)\cos(\phi)\,t, \quad h + v_o \sin(\phi)\,t - \frac{1}{2} g\,t^2 \right\rangle \\
&\approx \left\langle 61.2\,t,\ 61.2\,t,\ 60 + 50\,t - 16\,t^2 \right\rangle. \\
&\quad \text{for} \quad t \in [0, t^*]
\end{aligned}$$

5c To find where the ball lands in the xy-plane we just evaluate $x(t)$ and $y(t)$ at $t = t^*$.
It lands at $(x(t^*),\ y(t^*),\ 0)$ where

$$x(t^*) \approx 248.1, \quad \text{and} \quad y(t*) \approx 248.1$$

5d

$$\text{total distance in } xy\text{-plane} \ = \ \sqrt{[x(t^*)]^2 + [y(t^*)]^2} \approx 350.8 \text{ feet.}$$

5e The maximum height is achieved at $t = \tilde{t}$:

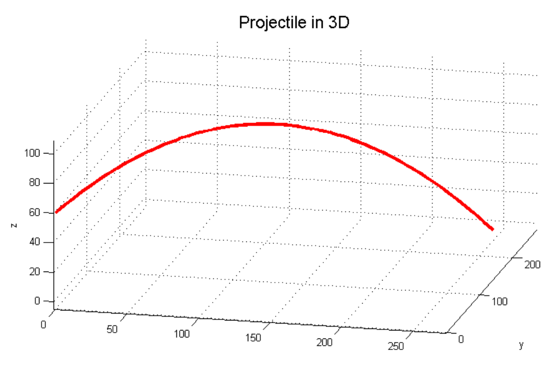
$$\tilde{t} = \frac{v_o \sin(\phi)}{g} \approx 1.56 \text{ seconds}$$

and the maximum height is $z(\tilde{t})$

$$z(\tilde{t}) \approx 99.1 \text{ feet.}$$

5f Below is the code and below that is a plot of the trajectory.

```
%% Homework for Chapter 3.3 #5
clc; clf; clear;

%% Initial Values
h = 60; phidegrees=30; thetadegrees=45; vo=100; g = 32;
theta = thetadegrees*pi/180; phi = phidegrees*pi/180;

%% Calculate Critical Values
tstar = (vo*sin(phi) + sqrt(vo^2 * (sin(phi))^2 + 2*g*h))/g;
fprintf('Hits the ground at t = %1.2f seconds. \n', tstar);
xmax = vo * cos(theta) * cos(phi) * tstar;
ymax = vo * sin(theta) * cos(phi) * tstar;
fprintf('It lands at (%1.1f, %1.1f, 0) \n', xmax, ymax);
distance = sqrt(xmax^2 + ymax^2);
fprintf('The total distance is %1.1f feet \n', distance);
ttilde = vo*sin(phi)/g;  % time of max heght
zmax = h + vo*sin(phi)*ttilde - 1/2*g*ttilde^2;  %max height
fprintf('The max height is %1.1f feet \n', zmax);

%%  Create Trajectory points
t = linspace(0,tstar,100);
x = vo * cos(theta) * cos(phi) * t;
y = vo * sin(theta) * cos(phi) * t;
z = h + vo*sin(phi)*t - 1/2*g*t.^2;

%% Plotting Directives
xyzbounds = [-1.5, 1.1*xmax, -1.5, 1.1*ymax, -5, 1.1*zmax];
%these are graph bounds: [xmin,xmax,ymin,ymax,zmin,zmax]

%% Plot Trajectory
plot3(x,y,z,'r-','linewidth',3);  % parametric plot in 3D.
axis(xyzbounds); view(13,32); grid on;
xlabel('x'); ylabel('y'); zlabel('z');
title('Projectile in 3D','fontsize',16)
```



7a Here, $h = 240$, $\phi = 30^o = \pi/6$, $\theta = 45^o = \pi/4$, $v_o = 64$ ft/sec, and $g = 32$ feet/sec^2.
Note: $\sin(\pi/4) = \cos(\pi/4) = \frac{\sqrt{2}}{2}$, $\cos(\pi/6) = \frac{\sqrt{3}}{2}$, $\sin(\pi/3) = \frac{1}{2}$
Plugging these values into the equation for when the ball hits the ground.

$$t^* = \frac{v_o \sin(\phi) + \sqrt{v_o^2 \sin^2(\phi) + 2gh}}{g}$$
$$t^* = \frac{64(1/2) + \sqrt{(64 * (1/2))^2 + 2 \cdot 32 \cdot 240}}{32} = \frac{32 + 128}{32} = 5 \text{ seconds}$$

7b $\mathbf{r}(t) = \langle\, x(t),\ y(t),\ z(t)\,\rangle$
$$\mathbf{r}(t) = \left\langle v_o \cos(\theta)\cos(\phi)\, t, \quad v_o \sin(\theta)\cos(\phi)\, t, \quad h + v_o \sin(\phi)\, t - \frac{1}{2} g\, t^2 \right\rangle$$
$$\mathbf{r}(t) = \left\langle 64 \frac{\sqrt{2}}{2}\frac{\sqrt{3}}{2} t, \quad 64 \frac{\sqrt{2}}{2}\frac{\sqrt{3}}{2} t, \quad 240 + 64\frac{1}{2}\, t - 16\, t^2 \right\rangle \text{ for } \quad t \in [0, t^*]$$
$$\mathbf{r}(t) = \left\langle 16\sqrt{6}\, t, \quad 16\sqrt{6}\, t, \quad 240 + 32\, t - 16\, t^2 \right\rangle \text{ for } \quad t \in [0, t^*]$$
To find where the ball lands in the xy-plane we just evaluate $x(t)$ and $y(t)$ at $t = t^* = 5$.
$\mathbf{r}(t^*) = \langle 80\sqrt{6}, \quad 80\sqrt{6}, \quad 240 + 160 - 16 \cdot 25 \rangle \approx \langle 196.0, \quad 196.0, \quad 0 \rangle$

7c $\mathbf{v}(t) = \langle v_o \cos(\theta)\cos(\phi),\ v_o \sin(\theta)\cos(\phi),\ v_o \sin(\phi) - g\, t \rangle$

$$\mathbf{v}(t) = \left\langle 16\sqrt{6}, \quad 16\sqrt{6}, \quad 32\ - 32\, t \right\rangle \text{ for } \quad t \in [0, t^*]$$
$$\mathbf{v}(t^*) = \mathbf{v}(5) = \left\langle 16\sqrt{6}, \quad 16\sqrt{6}, \quad -128 \right\rangle \approx \langle 39.2, 39.2, -128 \rangle$$

7d The speed is just the magnitude of the velocity vector at t^*.
$||\mathbf{v}(t^*)|| = \sqrt{(16\sqrt{6})^2 + (16\sqrt{6})^2 + (-128)^2}$
$||\mathbf{v}(t^*)|| = \sqrt{1536 + 1536 + 16,384} \approx 139.5$ ft/sec.

8a Here, $h = 60$, $\phi = 30^o = \pi/6$, $\theta = 45^o = \pi/4$, $v_o = 100$ ft/sec, and $g = 32$ feet/sec^2.
First determine when the ball hits the ground. You did this in problem 5 using equation (3.24),

$$t^* = \frac{v_o \sin(\phi) + \sqrt{v_o^2 \sin^2(\phi) + 2gh}}{g} \approx 4.05 \text{ seconds}$$

The velocity vector is found by differentiating the position vector:

$$\begin{aligned}
\mathbf{r}(t) &= \left\langle v_o \cos(\theta)\cos(\phi)\, t, \quad v_o \sin(\theta)\cos(\phi)\, t, \quad h + v_o \sin(\phi)\, t - \frac{1}{2} g\, t^2 \right\rangle \\
\mathbf{v}(t) &= \langle v_o \cos(\theta)\cos(\phi), \quad v_o \sin(\theta)\cos(\phi), \quad v_o \sin(\phi) - g\, t \rangle \\
&\approx \langle 61.2, \quad 61.2, \quad 50 - 32\, t \rangle\,.
\end{aligned}$$

So the velocity vector when it hits the ground is just $\mathbf{v}(t^*)$

$$\begin{aligned}
\mathbf{v}(t^*) &= \langle 61.2, \quad 61.2, \quad 50 - 32\, t^* \rangle \\
&\approx \langle 61.2, 61.2, -79.6 \rangle
\end{aligned}$$

8b The speed when it hits the ground is given by $||\mathbf{v}(t^*)||$.

$$||\mathbf{v}(t^*)|| = ||\, \langle 61.2, 61.2, -79.6 \rangle\, || \approx 117.64 \text{ feet/sec}$$

8c A perfect bounce is given by the formula in equation (3.31)

$$\mathbf{v}_1 = \mathbf{v}_0 - 2\,\mathrm{Proj}_{\mathbf{n}}\mathbf{v}_0 = \langle 61.2, 61.2, 79.6\rangle$$

where $\mathbf{v}_0$ is the incoming velocity (found in part (a) $= \mathbf{v}(t^*)$), n is the normal to the ground $= \langle 0, 0, 1\rangle$, and $\mathbf{v}_1$ is the bounce vector. Notice, for now, all the perfect bounce does is change the sign of the z component of the incoming velocity vector.

Below is the code used to answer these questions.

```
%% Homework for Chapter 3.3 #8
clc; clf; clear;

%% Initial Values
h = 60; phidegrees = 30; thetadegrees = 45; vo = 100; g = 32;
theta = thetadegrees*pi/180; phi = phidegrees*pi/180;

%% Calculate Critical Values
tstar = (vo*sin(phi) + sqrt(vo^2 * (sin(phi))^2 + 2*g*h))/g;
fprintf('It hits at t = %1.2f sec \n',tstar)

%% Derivative Functions
  dx = @(t,phi,theta,g)[vo * cos(theta) * cos(phi)] ;    % x'(t)
  dy = @(t,phi,theta,g)[vo * sin(theta) * cos(phi)] ;    % y'(t)
  dz = @(t,phi,theta,g)[vo*sin(phi) - g*t];     % z'(t)

%% Derivatives
dxstar = dx(tstar,phi,theta,g);   %x'(tstar)
dystar = dy(tstar,phi,theta,g);   %y'(tstar)
dzstar = dz(tstar,phi,theta,g);   %z'(tstar)
vstar = [dxstar, dystar, dzstar] %v(tstar)
impact_speed = norm(vstar)

Bounce_Vector = vstar - 2*ProjUV(vstar,[0,0,1])
```

Chapter 3.4

1a Using equation (3.13) (page 130) from Section 3.3 for the distance traveled,

$$\text{total distance} = \frac{v_o^2\cos(\theta)}{g}\left(\sin(\theta) + \sqrt{\sin^2(\theta) + \frac{2gh}{v_o^2}}\right)$$

with $\theta = \pi/6$, $v_o = 80$, $g = 32$, and $h = 10$, you should get **189.07** feet.

1b With $\Delta t = 0.10$, the ball hits the ground at $x \approx 200.9$ feet.

1e If you use $\Delta t = 0.001$ you should get a landing point of $x \approx 189.14$ which is within 0.1 of the actual landing position. It takes a long time for the program to run with such a small step size.

Chapter 3.5

3 The code from `Euler_2D_bounce_Wall.m` remains essentially the same with the prescribed differences in v_o, θ, D, and Δt. The trick is in the triangle collision detection and bounce response. Here, $W = (100, 0)$ and $W_1 = (160, 40)$ producing $\mathbf{w} = \langle 60, 40 \rangle$, which are easily obtained from the graph. It's all algebra from here to get the collision detection. We want to check if R_1 is in the triangle. So, the x-value has to be between 100 and 160 but the y-value must be below the line connecting W and W_1. That line is defined by $(y - 0) = m(x - 100)$ where $m = \frac{40}{60}$. So, you need to check if $R(2) < m(x - 100)$. If it is (while $R(1)$ is between 100 and 160), you've got a hit. Below is the code

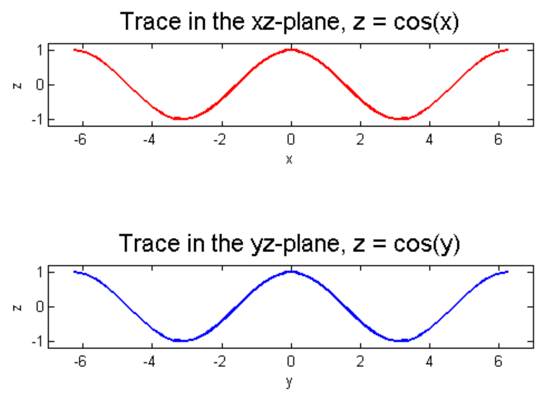
```
%% This is HW_3_5_3.m
% Performs a bouncing ball simulation with a non-horizontal wall
clf; clc; clear;
%% Initial Values
vo = 80; thetadegrees = 20; h=0; g = 32;
t_end = 9; delta_t = .05; t_steps = t_end/delta_t;
R=[0,h];   %/ initial position.
theta = thetadegrees*pi/180;
accel = [0,-g];   %/ the constant acceleration vector.
D = .8;   %/ D = damping on the bounce (between 0 and 1)
%% Wall
W_x_values = [100,160,160]; W_y_values = [0,40,0];
W=[100,0]; W1=[160,40]; w=(W1-W); slopeW = w(2)/w(1);

%% Plot first Point and  Wall
plot(R(1),R(2),'r.'); hold on;
plot(W_x_values,W_y_values,'k-','linewidth',4);
hold on; axis equal; axis([0,200,0,100]); grid on;
title('click to start','fontsize',16);
waitforbuttonpress % waits for a click on the graph.
title('Euler Bounce with Wall','fontsize',16);
%% Animation Loop
v = vo*[cos(theta), sin(theta)];   % initial velocity vector
for i = 1:1:t_steps
  V = delta_t*v;
  R1 = R + V;          % Next Position
  v1 = v + delta_t*accel;   % Next Velocity Vector
  if R1(2)<0                              % Detect ground hit
      disp('hit ground');
      V1 = D*(2*ProjUV(V,[1,0]) - V);     % New V1
      R1 = [R(1) - R(2)* V(1)/V(2), 0];   % New R1
      v1 = V1/delta_t;                    % New v1
  end
  if R1(1) >= W(1) & R1(1) <= W1(1) & R1(2) <= W(2)+slopeW*(R1(1)-W(1))
      disp('hit wall')                    % detect wall hit
      V1 = D*(2*ProjUV(V,w) - V);         % New V1
      rs=[-w(1),V(1);-w(2),V(2)]\[W(1)- R(1); W(2) - R(2)];
      R1 = R + rs(2)*V;                   % New R1
      v1 = V1/delta_t;                    % New v1
  end
  pause(.005);
  plot(R1(1),R1(2), 'r.','markersize',20);   %% plot the new point.
  R = R1; v = v1;   % Reset R and v.
end
```

Chapter 4.1

1a In the xz plane, $y = 0$, and z is given by $z = \cos\left(\sqrt{x^2}\right) = \cos|x| = \cos(x)$

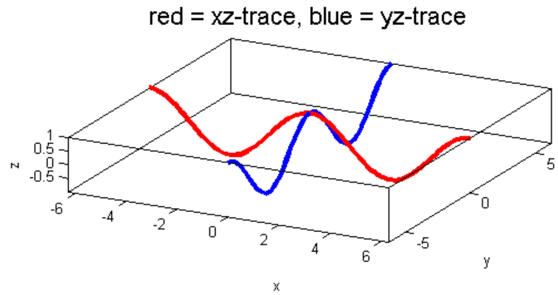
1b In the yz plane, $x = 0$, and z is given by $z = \cos\left(\sqrt{y^2}\right) = \cos|y| = \cos(y)$

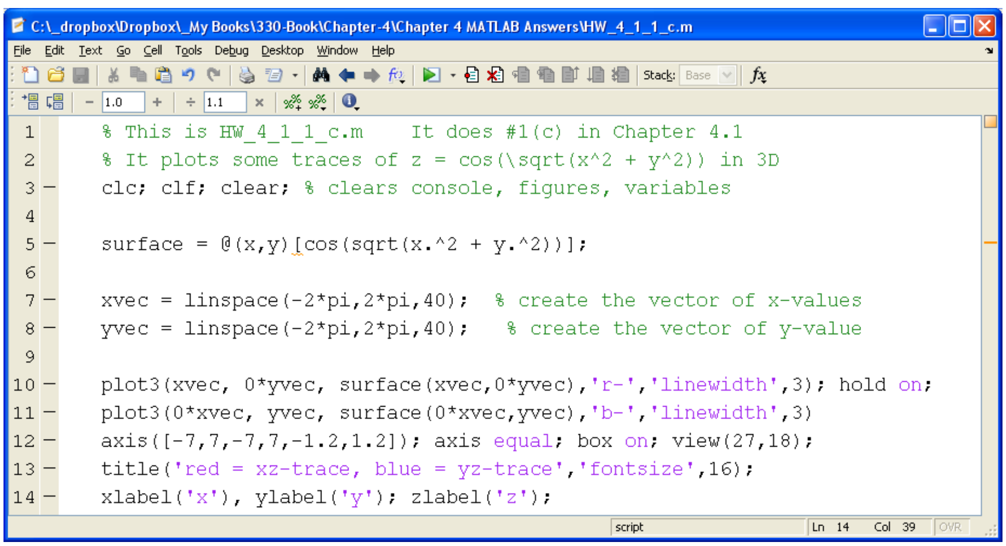
Below are the graphs of the traces and the code I used to make this graph.




```
%% This is HW4-1-1-ab.m    It does #1 (a) and (b) in Chapter 4.1
% It plots some traces of z = cos(\sqrt(x^2 + y^2))
clc; clf; clear; % clears console, figures, variables

%% The surface function
surface = @(x,y)[cos(sqrt(x.^2 + y.^2))];

%% The x and y values
xvec = linspace(-2*pi,2*pi,40);   % create the vector of x-values
yvec = linspace(-2*pi,2*pi,40);   % create the vector of y-value

%% Plotting Traces
%  In the xz plane, y = 0.
subplot(2,1,1)
plot(xvec,surface(xvec,0*yvec),'r-', 'linewidth',2)
axis equal; axis([-7,7,-1.2,1.2]);
title('Trace in the xz-plane, z = cos(x)','fontsize',16), xlabel('x'), ylabel('z')
%   In the yz plane, x = 0.
subplot(2,1,2)
plot(yvec,surface(0*xvec,yvec),'b-','linewidth',2)
axis equal; axis([-7,7,-1.2,1.2]);
title('Trace in the yz-plane, z = cos(y)','fontsize',16), xlabel('y'), ylabel('z')
```

1c Below is the graph and below that the code.






```
% This is HW_4_1_1_c.m    It does #1(c) in Chapter 4.1
% It plots some traces of z = cos(\sqrt(x^2 + y^2)) in 3D
clc; clf; clear; % clears console, figures, variables

surface = @(x,y)[cos(sqrt(x.^2 + y.^2))];

xvec = linspace(-2*pi,2*pi,40);  % create the vector of x-values
yvec = linspace(-2*pi,2*pi,40);   % create the vector of y-value

plot3(xvec, 0*yvec, surface(xvec,0*yvec),'r-','linewidth',3); hold on;
plot3(0*xvec, yvec, surface(0*xvec,yvec),'b-','linewidth',3)
axis([-7,7,-7,7,-1.2,1.2]); axis equal; box on; view(27,18);
title('red = xz-trace, blue = yz-trace','fontsize',16);
xlabel('x'), ylabel('y'); zlabel('z');
```

1(d)i If $z = \cos\left(\sqrt{x^2+y^2}\right)$ and $z = 0$ then $\cos\left(\sqrt{x^2+y^2}\right) = 0$ or

$$\sqrt{x^2+y^2} = \frac{\text{any odd integer}}{2} \cdot \pi = \frac{2k+1}{2}\,\pi \quad \text{for any integer } k$$

This results in concentric circles

$$x^2+y^2 = \left(\frac{2k+1}{2}\pi\right)^2$$

which are circles with radius $= \left|\dfrac{2k+1}{2}\pi\right|$.

1(d)ii If $z = \cos\left(\sqrt{x^2+y^2}\right)$ and $z = -1$ then $\cos\left(\sqrt{x^2+y^2}\right) = -1$ or

$$\sqrt{x^2+y^2} = (\text{any odd integer}) \cdot \pi = (2\,k+1)\,\pi \quad \text{for any integer } k$$

This results in concentric circles

$$x^2 + y^2 = [(2k+1)\pi]^2$$

which are circles with radius $= |(2k+1)\pi|$.

1(d)iii If $z = \cos\left(\sqrt{x^2+y^2}\right)$ and $z = 1$ then $\cos\left(\sqrt{x^2+y^2}\right) = 1$ or

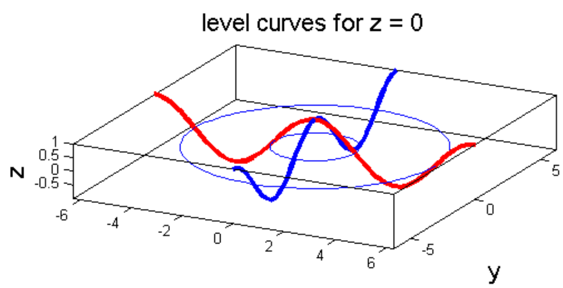
$$\sqrt{x^2+y^2} = (\text{any even integer}) \cdot \pi = (2\ k)\ \pi \quad \text{for any integer } k$$

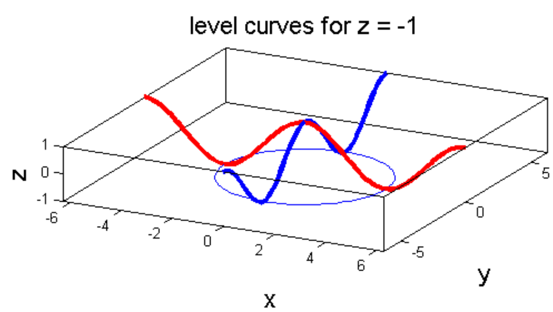
This results in concentric circles

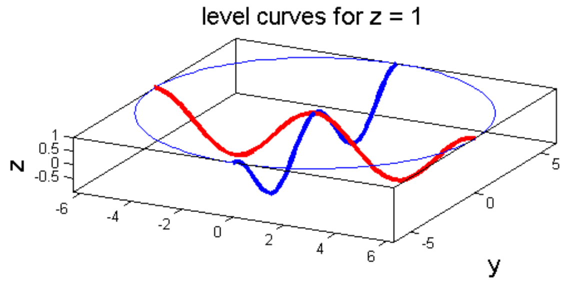
$$x^2 + y^2 = [2k\ \pi]^2$$

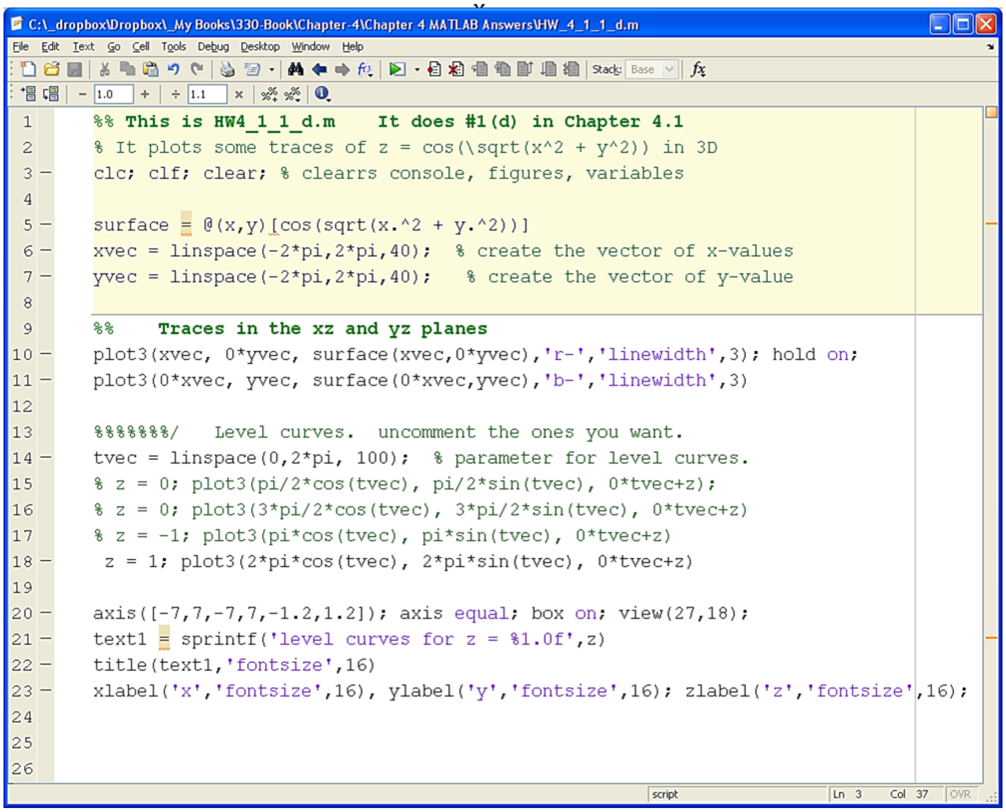
which are circles with radius $= |2k\ \pi|$.

Below are the level-curves in 3-space with the traces from parts (a) and (b).



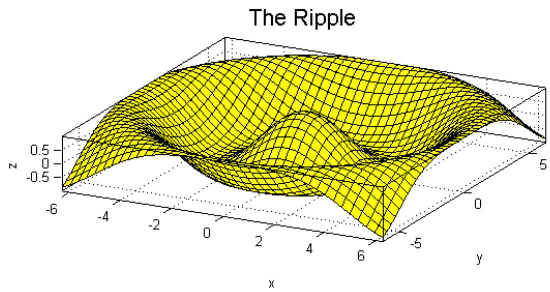







```
%% This is HW4_1_1_d.m    It does #1(d) in Chapter 4.1
% It plots some traces of z = cos(\sqrt(x^2 + y^2)) in 3D
clc; clf; clear; % clearrs console, figures, variables

surface = @(x,y)[cos(sqrt(x.^2 + y.^2))]
xvec = linspace(-2*pi,2*pi,40);  % create the vector of x-values
yvec = linspace(-2*pi,2*pi,40);   % create the vector of y-value

%%    Traces in the xz and yz planes
plot3(xvec, 0*yvec, surface(xvec,0*yvec),'r-','linewidth',3); hold on;
plot3(0*xvec, yvec, surface(0*xvec,yvec),'b-','linewidth',3)

%%%%%%%/   Level curves.  uncomment the ones you want.
tvec = linspace(0,2*pi, 100);  % parameter for level curves.
% z = 0; plot3(pi/2*cos(tvec), pi/2*sin(tvec), 0*tvec+z);
% z = 0; plot3(3*pi/2*cos(tvec), 3*pi/2*sin(tvec), 0*tvec+z)
% z = -1; plot3(pi*cos(tvec), pi*sin(tvec), 0*tvec+z)
 z = 1; plot3(2*pi*cos(tvec), 2*pi*sin(tvec), 0*tvec+z)

axis([-7,7,-7,7,-1.2,1.2]); axis equal; box on; view(27,18);
text1 = sprintf('level curves for z = %1.0f',z)
title(text1,'fontsize',16)
xlabel('x','fontsize',16), ylabel('y','fontsize',16); zlabel('z','fontsize',16);
```

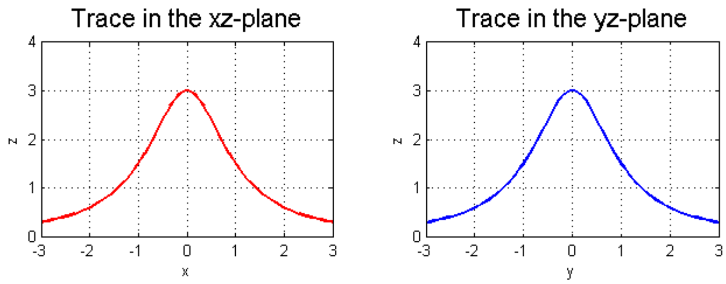
1e Below is the graph of the surface. Below that is the code.




```
% This is HW_4_1_1_e.m    It does #1(e) in Chapter 4.1
% It plots some traces of z = cos(\sqrt(x^2 + y^2)) in 3D
clc; clf; clear; % clearrs console, figures, variables

surface = @(x,y)[cos(sqrt(x.^2 + y.^2))]
xvec = linspace(-2*pi,2*pi,40);  % vector of x-values
yvec = linspace(-2*pi,2*pi,40);   % vector of y-value

% Plot z = x^2 + y^2 + 2
[x y] = meshgrid(xvec,yvec);  % create a full array of x & y
z = surface(x,y);  % define z
surf(x, y, z,'facecol', 'yellow');
title('The Ripple','fontsize',16)
axis([-7,7,-7,7,-1.2,1.2]); axis equal; box on; view(27,18);
xlabel('x'); ylabel('y');  zlabel('z');
```

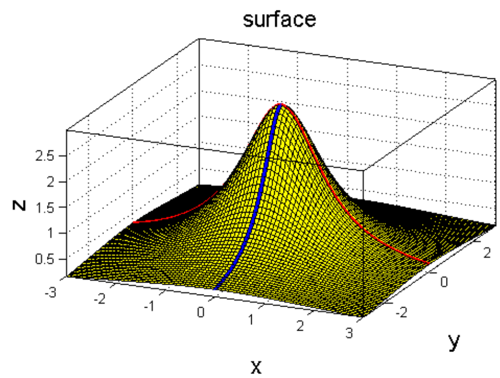
3a In the xz-plane, $y = 0$, and z is given by $z = \dfrac{3}{1+x^2}$

In the yz-plane, $x = 0$, and z is given by $z = \dfrac{3}{1+y^2}$



3b The surface:

$$f(x, y) = \frac{3}{1 + x^2 + y^2}$$



5a Solving the equation for z we get,

$$z = f(x, y) = \frac{3}{4}x + \frac{1}{2}y - \frac{3}{2}$$

5b This can be done in two ways:

$$F(x, y, z) = 3x + 2y - 4z - 6 \qquad \text{or} \qquad F(x, y, z) = -3x - 2y + 4z + 6$$

Either way, the equation $F(x, y, z) = 0$ represents the equation for the same plane.

Chapter 4.2

1a $f_x = y + 2x$ and $f_y = x + 2y$

1c f_x requires the product rule treating y as a constant.
$f_x = y\,[x(-\sin(2x)\;2) + \cos(2x)] = y\,[-2x\sin(2x) + cos(2x)]$
$f_y = x\cos(2x)$

1e $F_x = 3y^2z^3$, $F_y = 6xyz^3$, and $F_z = 9xy^2z^2$

2a We found f_x and f_y in Example 3 at the beginning of this chapter.

$$f_x = \frac{-x\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}} \qquad \text{and} \qquad f_y = \frac{-y\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$$

2b The normal vector to the surface is either

$$\mathbf{n} = \langle f_x, f_y, -1 \rangle \quad \text{or} \quad \langle -f_x, -f_y, 1 \rangle$$

If we want the normal vector that points up, we choose the second form with positive z-component so

$$\mathbf{n} = \langle -f_x, -f_y, 1 \rangle = \left\langle \frac{x \sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}, \frac{y \sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}, 1 \right\rangle$$

Now we just evaluate this at the point $(x, y) = (1, 0)$. or

$$\mathbf{n} = \langle\ \sin(1),\ 0,\ 1\ \rangle \approx \langle\ 0.8415,\ 0,\ 1\ \rangle$$

So, the proper **unit** normal vector is then

$$\mathbf{u} = \frac{n}{||n||} \approx \langle 0.6439,\ 0,\ 0.7652 \rangle$$

2c Below is the code used to generate the graph and unit normal vector.

```
%% This is HW_4_2_2.m It does #2 from Chapter 4.2
% This plots the ripple surface with a normal vector
clc; clf; clear;

%% The surface function and partial derivatives.
f = @(x,y)[cos(sqrt(x.^2 + y.^2))];  %The ripple function
  fx = @(x,y)[-x*sin(sqrt(x^2 + y^2))/sqrt(x^2 + y^2)];
  fy = @(x,y)[-y*sin(sqrt(x^2 + y^2))/sqrt(x^2 + y^2)];

%% The Normal Vectors at P
P = [1,0,f(1,0)];
n_up = [-fx(P(1),P(2)),-fy(P(1),P(2)),1];  %upward normal
n_down = [fx(P(1),P(2)),fy(P(1),P(2)),-1]; %downward normal
n = n_up;   %choose the upward normal
u = n/norm(n); %normalize the normal vector (make length = 1)

%% The Plot
xvec = linspace(-2*pi,2*pi,40);  % creates the vector of x-values
yvec = linspace(-2*pi,2*pi,40);  % creates the vector of y-values
[x y] = meshgrid(xvec,yvec);     % creates a full array of x & y values
z = f(x,y);                      % creates the full array of z values
surf(x, y, z,'facecol', 'yellow');  hold on; % creates the surface plot
vectarrow(P,P+n,'black'); % creates the normal vector
plot3(P(1),P(2),P(3), 'k.','markersize',40) % plot the point
axis equal; axis([-7,7,-7,7,-1.2,2]); box on; view(27,18);
title('Surface and Normal Vector','fontsize',16)
xlabel('x','fontsize',16);ylabel('y','fontsize',16);zlabel('z','fontsize',16);
```

4a $f_x = -3\left(1 + x^2 + y^2\right)^{-2}\ 2x\ = \dfrac{-6x}{(1 + x^2 + y^2)^2}$

$f_y = -3\left(1 + x^2 + y^2\right)^{-2}\ 2y\ = \dfrac{-6y}{(1 + x^2 + y^2)^2}$

4b $\mathbf{n} = \langle f_x, f_y, -1\rangle$ or $\mathbf{n} = \langle -f_x, -f_y, 1\rangle$. We choose the second one because it points up and out.
$f_x(-1,1) = \dfrac{6}{(1+1+1)^2} = \dfrac{6}{9} = \dfrac{2}{3}$ so $-f_x(-1,1) = -\frac{2}{3}$
$f_y(-1,1) = \dfrac{-6}{(1+1+1)^2} = \dfrac{-6}{9} = \dfrac{-2}{3}$ so $-f_x(-1,1) = \frac{2}{3}$
$\mathbf{n} = \left\langle -\frac{2}{3}, \frac{2}{3}, 1\right\rangle \approx \langle -0.67, 0.67, 1\rangle$
So, the proper **unit** normal vector is then

$$\mathbf{u} = \frac{n}{||n||} \approx \langle -0.4851,\ 0.4851,\ 0.7276\rangle$$

6a $\nabla F = \langle F_x,\ F_y,\ F_z\rangle = \langle\ 6x,\ -2,\ -3\cos(z)\ \rangle$

8a When you solve for z you get
$z = f(x,y) = -2 \pm \sqrt{25-(x-2)^2-(y-1)^2}$. Since our point (2,4,-6) is on the bottom half of the sphere, we choose the minus sign and

$$\begin{aligned} z = f(x,y) &= -2 - \sqrt{25-(x-2)^2-(y-1)^2} \\ &= -2 - \left(25-(x-2)^2-(y-1)^2\right)^{1/2} \end{aligned}$$

Now, using the chain rule with the power rule, the partial derivatives f_x and f_y are given by

$$\begin{aligned} f_x &= -\frac{1}{2}(25-(x-2)^2-(y-1)^2)^{-1/2}(-2(x-2)) = \frac{x-2}{\sqrt{25-(x-2)^2-(y-1)^2}} \\ f_y &= -\frac{1}{2}(25-(x-2)^2-(y-1)^2)^{-1/2}(-2(y-1)) = \frac{y-1}{\sqrt{25-(x-2)^2-(y-1)^2}}. \end{aligned}$$

Now, evaluating these at $x = 2$, and $y = 4$,

$$\begin{aligned} f_x(2,4) &= \frac{2-2}{\sqrt{25-(2-2)^2-(4-1)^2}} = 0 \\ f_y(2,4) &= \frac{4-1}{\sqrt{25-(2-2)^2-(4-1)^2}} = \frac{3}{4}. \end{aligned}$$

and

$$\mathbf{n} = \pm\langle f_x, f_y, -1\rangle = \pm\left\langle\ 0,\ \frac{3}{4},\ -1\ \right\rangle \quad \text{choose } (+) \rightarrow \quad \mathbf{n} = \left\langle\ 0,\ \frac{3}{4},\ -1\ \right\rangle$$

Since our point is on the bottom half of the sphere, the outward-pointing normal will have a negative z-component so you choose the positive (+) version of these normal vectors.

8b Taking the equation for the sphere and getting a zero on the right hand side:

$$F(x,y,z) = (x-2)^2 + (y-1)^2 + (z+2)^2 - 25$$

then

$$F_x = 2(x-2) \qquad F_y = 2(y-1) \qquad F_z = 2(z+2)$$

and plugging in $x = 2$, $y = 4$, and $z = -6$ yields

$$F_x = 2(2-2) = 0 \qquad F_y = 2(4-1) = 6 \qquad F_z = 2(-6+2) = -8$$

and

$$\mathbf{n} = \pm\nabla F = \pm\langle 0, 6, -8\rangle.$$

Again, since our point (2,4,-6) is on the bottom of the sphere we choose the (+) version and $\mathbf{n} = \langle 0, 6, -8\rangle$. Notice this is parallel to the normal found in part (a) and was a lot easier to obtain.

Chapter 4.3

1 In the previous chapter, we found the normal vector to the surface at $(1, 0, \cos(1))$.
$\mathbf{n} = \langle\ \sin(1),\ 0,\ 1\ \rangle \approx \langle\ 0.84,\ 0,\ 1\ \rangle$
Now we use the equation: $\mathbf{v}_1 = \mathbf{v}_o - 2\text{Proj}_{\mathbf{n}}\mathbf{v}_o$, where

$\text{Proj}_{\mathbf{n}}\mathbf{v}_o = \frac{\mathbf{n}\cdot\mathbf{v}_o}{||\mathbf{n}||^2}\mathbf{n} = \frac{\langle\sin(1),0,1\rangle\cdot\langle 0,0,-3\rangle}{\sin^2(1)+1}\langle\sin(1), 0, 1\rangle = \frac{-3}{\sin^2(1)+1}\langle\sin(1), 0, 1\rangle$

Now, $\mathbf{v}_1 = \mathbf{v}_o - 2\text{Proj}_{\mathbf{n}}\mathbf{v}_o = \langle 0, 0, -3\rangle + \frac{6}{\sin^2(1)+1}\langle\sin(1), 0, 1\rangle \approx \langle\mathbf{2.9559},\ \mathbf{0.000},\ \mathbf{0.5127}\rangle$

3 In the previous chapter, we found the normal vector to the surface at $(-1, 1, 1)$.
$\mathbf{n} = \left\langle -\frac{2}{3}, \frac{2}{3}, 1\right\rangle \approx \langle -0.67, 0.67, 1\rangle.$
Now we use the equation: $\mathbf{v}_1 = \mathbf{v}_o - 2\text{Proj}_{\mathbf{n}}\mathbf{v}_o$, where

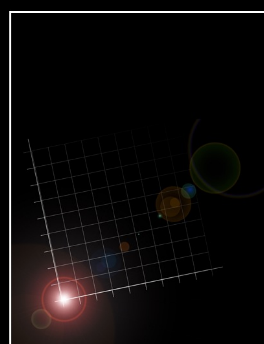
$\text{Proj}_{\mathbf{n}}\mathbf{v}_o = \frac{\mathbf{n}\cdot\mathbf{v}_o}{||\mathbf{n}||^2}\mathbf{n} = \frac{\left\langle -\frac{2}{3},\frac{2}{3},1\right\rangle\cdot\langle 0,-1,-1\rangle}{17/9}\left\langle -\frac{2}{3}, \frac{2}{3}, 1\right\rangle = \frac{-5/3}{17/9}\left\langle -\frac{2}{3}, \frac{2}{3}, 1\right\rangle = \frac{-15}{17}\left\langle -\frac{2}{3}, \frac{2}{3}, 1\right\rangle = \frac{-5}{17}\langle -2, 2, 3\rangle$

Now, $\mathbf{v}_1 = \mathbf{v}_o - 2\text{Proj}_{\mathbf{n}}\mathbf{v}_o = \langle 0, -1, -1\rangle + \frac{10}{17}\langle -2, 2, 3\rangle = \left\langle \frac{-20}{17}, \frac{3}{17}, \frac{13}{17}\right\rangle \approx \langle\mathbf{-1.1765},\ \mathbf{0.1765},\ \mathbf{0.7647}\rangle$

5 $\mathbf{v}_1 = \mathbf{v}_0 - 2\text{Proj}_n\mathbf{v}_0$, where $\mathbf{n}$ is a normal to the plane. $\mathbf{n} = \langle 2, 2, -1\rangle$.
$\mathbf{v}_1 = \mathbf{v}_0 - 2 * \frac{\mathbf{n}\cdot\mathbf{v}_0}{||\mathbf{n}||^2}\ \mathbf{n} = \langle 2, 3, 6\rangle - 2\frac{\mathbf{n}\cdot\mathbf{v}_0}{||\mathbf{n}||^2}\langle 2, 2, -1\rangle = \langle 2, 3, 6\rangle - 2\frac{\langle 2,2,-1\rangle\cdot\langle 2,3,6\rangle}{9}\langle 2, 2, -1\rangle$
$\mathbf{v_1} = \langle 2, 3, 6\rangle - 2\frac{4}{9}\langle 2, 2, -1\rangle = \langle 2, 3, 6\rangle - \frac{8}{9}\langle 2, 2, -1\rangle = \left\langle \frac{18}{9}, \frac{27}{9}, \frac{54}{9}\right\rangle - \left\langle \frac{16}{9}, \frac{16}{9}, \frac{-8}{9}\right\rangle$
$\mathbf{v}_1 = \left\langle \mathbf{\frac{2}{9}}, \mathbf{\frac{11}{9}}, \mathbf{\frac{62}{9}}\right\rangle \approx \langle\mathbf{0.22},\ \mathbf{1.22},\ \mathbf{6.89}\rangle$

Index

Matrices, Vectors, and 3D Math



Worldwide Center of Mathematics, LLC
www.centerofmath.org